

User Registration System (URS)

API Reference Implementation

Version 2.56

EOSDIS User Registration System (URS)

Overview

The EOSDIS URS provides a centralized and simplified mechanism for user registration and account management for all EOSDIS system components. End users may register and edit their account information in one location allowing them access to the wide array of EOSDIS data and services. The EOSDIS URS also helps the EOSDIS program better understand the user demographics and access patterns in support of planning for new value-added features and customized services that can be directed to specific users or user groups resulting in better user experience.

For questions regarding the EOSDIS URS, please contact support at support@earthdata.nasa.gov.

User Migrations

Many EOSDIS system components have an existing user registration and management system. Over time, these disparate user registration systems will be consolidated and migrated to utilize the new EOSDIS URS. Each EOSDIS system component will schedule and execute its own user migration. Users of those systems will be contacted and notified regarding the appropriate migration steps. The EOSDIS URS enforces account uniqueness based on email address and user id. As a result, it is expected that account conflicts will arise as additional EOSDIS systems migrate their users. EOSDIS will work with each system component to ensure that users are presented with consistent and correct account migration information and procedures.

Requesting URS Access

URS is bringing on clients slowly and in a controlled manner to ensure the system is configured and scaled to meet all user requirements, and operate at peak efficiency. New clients will need to contact EOSDIS and the URS team to request access. Access will be provided based upon client technologies and requirements, EOSDIS prioritization, and current URS development and operational efforts. Requests for access can be sent to support@earthdata.nasa.gov.

Once you are granted access, you will be supplied a client application account and instructions on connecting to a specific URS testing authentication host. Further instructions about connecting to URS will be provided at that time.

Using this Document

This document is intended to be a developer guide to the publicly-exposed URS endpoints. These endpoints are available through HTTP calls, typically accomplished using a REST API like the built in REST support in [Ruby on Rails](#) or an API framework like [Restlet](#) for Java.

Selecting the Correct Service Endpoint

The naming of the Java Class that provides service is the first "clue" in selecting an endpoint for a given task. Following this section is an overview of the gov.nasa.eosdis.urs.api classes, and these are the high-level service containers. Each container is named in accordance with what that container does. GroupResource handles group-related functionality; LoginResource handles logging in; etc.

After locating the service container, simply look at the various endpoints. Each has a detailed description and sample request, as well as a fairly self-descriptive name. Once you've located the correct endpoint, you can construct a request using the sample HTTP request for each endpoint.

Using the Sample HTTP Requests

For each endpoint, an example curl request is shown. This request demonstrates the request body (typically an XML document fragment), any required or optional headers, and the URL for that endpoint. These sample requests can be run from the command-line by inserting the correct authentication host, port, and data. Using curl for a request is an excellent way to verify access to URS authentication hosts and endpoints.

Once connectivity and parameters are established, the request body, headers, and connection information should be converted to the language used for access (Rails, Java, etc.). Because the request will look different in every client language, and then again based upon the REST or HTTP client being used, it is up to the developer to convert the curl sample request into usable code on the client.

User Registration System API 2.55 API

Packages

[gov.nasa.eosdis.urs.api](#)

[gov.nasa.eosdis.urs.api.exceptions](#)

[gov.nasa.eosdis.urs.api.objects](#)

[gov.nasa.eosdis.urs.extendedwadl](#)

[gov.nasa.eosdis.urs.util](#)

[gov.nasa.eosdis.urs.webapp](#)

Copyright © 2012. All Rights Reserved.

Package gov.nasa.eosdis.urs.api

| Class Summary | |
|--------------------------------------|---|
| <u>GroupResource</u> | Server side REST interface for performing group operations. |
| <u>LoginResource</u> | Server side REST interface for performing user authentication operations. This class handles the logins to URS for other applications. |
| <u>MetricResource</u> | This class defines methods that can be used to retrieve new users created in LDAP and authentication metrics given a date time range. |
| <u>RpcResource</u> | Server side REST interface for performing RPC operations. |
| <u>TestResource</u> | Tools and utilities used only for testing purposes. This class is not deployed in the operational system. |
| <u>URSBaseResource</u> | The base class URSBaseResource is the super class of classes GroupResource, LoginResource, RpcResource, TestResource and UserResource. |
| <u>UserResource</u> | Server side REST interface for performing user based operations, such as create, read, update, and delete. |
| <u>UserValidationResource</u> | Server-side REST interface for validating various components of a user before creation of or updates to that user. |
| <u>ValidsResource</u> | Server side REST interface for requesting lists of valid values for a given property. |

Copyright © 2012. All Rights Reserved.

Package gov.nasa.eosdis.urs.api.exceptions

| Exception Summary | |
|------------------------------------|---|
| <u>BadRequestException</u> | |
| <u>ConflictErrorException</u> | |
| <u>ForbiddenException</u> | |
| <u>InternalServerError</u> | |
| <u>ServiceUnavailableException</u> | |
| <u>UnauthorizedException</u> | Exception to support basic http authentication. |
| <u>ValidationErrorsException</u> | |

Copyright © 2012. All Rights Reserved.

Package gov.nasa.eosdis.urs.api.objects

Class Summary

[GroupList](#)

[LoginRequest](#)

[ValidationErrorResponse](#)

[ValidationErrors](#)

Copyright © 2012. All Rights Reserved.

Package gov.nasa.eosdis.urs.extendedwadl

Class Summary

| | |
|-------------------------|---|
| URSModelExamples | This class provides example representations that can be used for wadl docs. |
|-------------------------|---|

Copyright © 2012. All Rights Reserved.

Package gov.nasa.eosdis.urs.util

| Class Summary | |
|--|--|
| <u>AccountVerifier</u> | |
| <u>AuthorityValidation</u> | |
| <u>GroupUtils</u> | |
| <u>Mailer</u> | Utility class used to compose emails for sending. |
| <u>PasswordUtils</u> | |
| <u>Randomizer</u> | Randomizer is a pseudo random number generator that builds on java.util.Random and allows multiple instances to share a single instance of that class, which should be more random than creating a new Random instance every time one is needed. |
| <u>ResourceLoader</u> | |
| <u>StringConstants</u> | |
| <u>UserUtils</u> | |
| <u>UserValidation</u> | |

Copyright © 2012. All Rights Reserved.

Package gov.nasa.eosdis.urs.webapp

Class Summary

[UrsApiServletContextListener](#)

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api Class GroupResource

```
java.lang.Object
└─gov.nasa.eosdis.urs.api.URSBaseResource
    └─gov.nasa.eosdis.urs.api.GroupResource
```

```
public class GroupResource
extends URSBaseResource
```

Server side REST interface for performing group operations.

Has methods that will create, add, update, and delete groups. Also can search and return all details for a group and a list of groups available for use.

Endpoints:

root-url - the path taken to reach the endpoint. For example, it can be thought of as:

<https://<authentication-host>:<port>/groups> or <https://<authentication-host>:<port>/groups>

<root-url>/groups

◊ This end point accepts two kinds of requests:

1. It accepts data submitted in XML form to create a group.
2. Returns a list of the groups available for use.

<root-url>/groups/<group>

◊ This end point accepts three kinds of requests:

1. Returns all the details for the named group.
2. Update group's users.
3. Deletes a group from the database.

<root-url>/groups/<group>/<user>

◊ This end point accepts two kinds of requests:

1. Adds a user to the named group.
2. Delete a user from the named group.

Additional HTML header parameters are

authentication

- (required) the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
-

Constructor Summary

[GroupResource \(\)](#)

Method Summary

Overview (User Registration System API 2.55 API)

| | | |
|--|------|---|
| | void | <u>addUserToGroup</u> (java.lang.String authorization, java.lang.String groupName, java.lang.String name) Adds a user to a group. |
| javax.ws.rs.core.Response | | <u>createGroup</u> (java.lang.String authorization, gov.nasa.eosdis.urs.core.objects.Group group) Creates a named group. |
| | void | <u>deleteGroup</u> (java.lang.String authorization, java.lang.String groupName) Deletes a group from the database. |
| | void | <u>deleteUserFromGroup</u> (java.lang.String authorization, java.lang.String groupName, java.lang.String userName) Delete user from group. |
| gov.nasa.eosdis.urs.core.objects.Group | | <u>getGroup</u> (java.lang.String authorization, java.lang.String groupName) Returns all the details for a named group. |
| <u>GroupList</u> | | <u>getGroups</u> (java.lang.String authorization) Returns a list of the groups available for use. |
| | void | <u>updateGroupUsers</u> (java.lang.String authorization, java.lang.String groupId, gov.nasa.eosdis.urs.core.objects.Group updatedGroup) Update group's users. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

GroupResource

```
public GroupResource()
```

Method Detail

getGroups

```
public GroupList getGroups(@HeaderParam(value="authorization")
                           java.lang.String authorization)
```

Returns a list of the groups available for use.

Example

Overview (User Registration System API 2.55 API)

Request list of groups

```
curl 'https://<authentication-host>:<port>/groups'  
--header "authorization:Basic <EncodedExampleAuthorization>"  
  
<authentication-host>  
    - The host that URS API is running on  
<port>  
    - The port number of the URS API  
<EncodedExampleAuthorization>  
    - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Returns all groups visible to the authority used |
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the get |
| 401 | Unauthorized | API authentication invalid |
| 500 | Internal Server Error | LDAP error, non-connectivity related |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)

Returns:

a list of names of groups. This may be restricted, depending upon the caller.

See Also:

GroupDAO

getGroup

```
public gov.nasa.eosdis.urs.core.objects.Group getGroup (@HeaderParam(value="authorization")  
                                         java.lang.String authorization,  
                                         java.lang.String groupName)
```

Returns all the details for a named group.

Example

Request details of a group

```
curl 'https://<authentication-host>:<port>/groups/<group>'  
--header "authorization:Basic <EncodedExampleAuthorization>"  
  
<authentication-host>  
    - The host that URS API is running on  
<port>  
    - The port number of the URS API  
<group>  
    - The group name.  
<EncodedExampleAuthorization>  
    - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---------------------------|
|-------------|------------------|---------------------------|

Overview (User Registration System API 2.55 API)

| | | |
|-----|-----------------------|---|
| 200 | OK | Returns single requested group if visible to the given authority |
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the get |
| 401 | Unauthorized | API authentication invalid |
| 500 | Internal Server Error | LDAP error, non-connectivity related |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)
 groupName - the name of the group to retrieve

Returns:

the named group. This includes a member and admin list, which may be incomplete, depending upon permissions.

See Also:

GroupDAO

createGroup

```
public javax.ws.rs.core.Response createGroup(@HeaderParam(value="authorization")
                                              java.lang.String authorization,
                                              gov.nasa.eosdis.urs.core.objects.Group group)
```

Creates a named group.

Examples:

XML encoded group creation

```
curl -d ' \
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
<group> \
    <description>ExampleDescription</description> \
    <name>ExampleName</name> \
</group> \
' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/xml" \
'https://<authentication-host>:<port>/groups'

<ExampleDescription>
    - The description of the group
<ExampleName>
    - The name of the group
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
Response
    a User object containing whole user information in XML format.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---------------------------|
| | | |

Overview (User Registration System API 2.55 API)

| | | |
|-----|-----------------------|--|
| 201 | OK | Group created successfully, the location header contains a URL that can be used to retrieve the group |
| 400 | Bad Request | Missing or invalid group name |
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the creation |
| 409 | Conflict | A group with this name already exists |
| 500 | Internal Server Error | LDAP error, non-connectivity related |
| | | The location URL for the new group is invalid. The group has been created, but something went wrong while constructing the response. |
| 401 | Unauthorized | API authentication invalid |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)
group - the group to be created

Returns:

the created group.

See Also:

[GroupDAO](#)

updateGroupUsers

```
public void updateGroupUsers (@HeaderParam(value="authorization")
                             java.lang.String authorization,
                             java.lang.String groupId,
                             gov.nasa.eosdis.urs.core.objects.Group updatedGroup)
```

Update group's users. The incoming group xml should have the entire list of users.

Example:

Update group request

```
curl -X PUT -d '
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
<group> \
  <members> \
    <member>UpdateMembers1</member> \
    <member>UpdateMembers2</member> \
  </members> \
</group> \
' \
--header "authorization:Basic <EncodedExampleAuthorization>" \
--header "content-type:application/xml" \
'https://<authentication-host>:<port>/groups/<group>' \
<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
<group>
```

Overview (User Registration System API 2.55 API)

- The name of the group to update.

<member>

- The member to be added to the group

<EncodedExampleAuthorization>

- The basic HTTP authentication header parameter.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|--|
| 204 | No Content | group's users are correctly updated, no other message returned |
| 400 | Bad Request | Group in LDAP contains bad data and cannot be properly retrieved |
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the creation |
| 404 | Not Found | The group being modified doesn't exist |
| 500 | Internal Server Error | LDAP error, non-connectivity related |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)

groupId - id of the group to be updated

updatedGroup - new group object

See Also:

GroupDAO

deleteGroup

```
public void deleteGroup(@HeaderParam(value="authorization")
                      java.lang.String authorization,
                      java.lang.String groupName)
```

Deletes a group from the database.

Example:

Delete group request

```
curl -X DELETE --header "authorization:Basic <EncodedExampleAuthorization>" \
      'https://<authentication-host>:<port>/groups/<group>'

<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<group>
    - The group to delete.
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 204 | No Content | group is deleted |
| 400 | Bad Request | Group in LDAP contains bad data and cannot be properly retrieved |

Overview (User Registration System API 2.55 API)

| | | |
|-----|-----------------------|--|
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the creation |
| 404 | Not Found | The group being deleted doesn't exist |
| 500 | Internal Server Error | LDAP error, non-connectivity related |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)
groupName - the name of the group to delete

See Also:

GroupDAO

deleteUserFromGroup

```
public void deleteUserFromGroup(@HeaderParam(value="authorization")
                                java.lang.String authorization,
                                java.lang.String groupName,
                                java.lang.String userName)
```

Delete user from group.

Example:

Request a user to be deleted from a group

```
curl -X DELETE --header "authorization:Basic <EncodedExampleAuthorization>" \
      'https://<authentication-host>:<port>/groups/<group>/<user>'

<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<uid>
    - The user to delete.
<group>
    - The group name.
<user>
    - The user to be deleted from the group.
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|--|
| 204 | No Content | user is deleted from group |
| 400 | Bad Request | Group in LDAP contains bad data and cannot be properly retrieved |
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the deletion |
| 404 | Not Found | user being deleted doesn't exist group being acted upon doesn't exist |
| 500 | Internal Server Error | LDAP error, non-connectivity related |

Overview (User Registration System API 2.55 API)

| | | |
|-----|---------------------|------------------------------------|
| | | LDAP search for user related error |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)
 groupName - the group name
 userName - the user name

addUserToGroup

```
public void addUserToGroup(@HeaderParam(value="authorization")
                           java.lang.String authorization,
                           java.lang.String groupName,
                           java.lang.String name)
```

Adds a user to a group.

Example:

Add a user to a group request

```
curl -X PUT -d '
      <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
      <group> \
          <members> \
              <member>AddedMember</member> \
          </members> \
      <group> \
      ' \
      --header "authorization:Basic <EncodedExampleAuthorization>" \
      --header "content-type: application/xml" \
      'https://<authentication-host>:<port>/groups/<group>/<user>' \
      <authentication-host>
          - The host that URS API is running on
      <port>
          - The port number of the URS API
      <user>
          - The name of the user to add to the group.
      <AddedMember>
          - Member to add to group
      <EncodedExampleAuthorization>
          - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---|
| 204 | No Content | user is added to group |
| 400 | Bad Request | Group in LDAP contains bad data and cannot be properly retrieved |
| 401 | Unauthorized | Invalid or missing authority |
| 403 | Forbidden | The authority provided, although valid, is not allowed to perform the add operation |
| 404 | Not Found | user being added doesn't exist group being acted upon doesn't exist |

Overview (User Registration System API 2.55 API)

| | | |
|-----|-----------------------|--|
| 500 | Internal Server Error | LDAP error, non-connectivity related LDAP search for user related error |
| 503 | Service Unavailable | LDAP connectivity related error |

Parameters:

authorization - basic HTTP authentication (http://en.wikipedia.org/wiki/Basic_access_authentication)

groupName - the group name

name - the user name or the email address

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api Class LoginResource

```
java.lang.Object
└─gov.nasa.eosdis.urs.api.URSBaseResource
    └─gov.nasa.eosdis.urs.api.LoginResource
```

```
public class LoginResource
extends URSBaseResource
```

Server side REST interface for performing user authentication operations.
This class handles the logins to URS for other applications.

Endpoints:

root-url - the path taken to reach the endpoint. For example, it can be thought of as:

<https://<authentication-host>:<port>/groups>

<<root-url>/login>

- ◊ All the login requests will be directed to this URL.
- ◊ This end point accepts two kinds of login requests:
 1. It handles passed-in LoginRequest (an XML object)
 2. It also accepts the data submitted in the form from a web GUI page
- ◊ Additional HTML header parameters are
 - authentication
 - (required) the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
 - X-On-Behalf-Of
 - (Optional) If provided, indicates that the authentication is being performed on behalf of an application (other than that identified by the credentials).

Constructor Summary

[LoginResource \(\)](#)

Method Summary

| | |
|--|--|
| gov.nasa.eosdis.urs.core.objects.LoginResponse | <p><u>postLogin</u>(java.lang.String authorization, java.lang.String onBehalfOf, <u>LoginRequest</u> request)</p> <p>POST method for performing a user authentication.</p> |
| gov.nasa.eosdis.urs.core.objects.LoginResponse | <p><u>postLoginUrlEncoded</u>(java.lang.String authorization, java.lang.String onBehalfOf, java.lang.String username,</p> |

Overview (User Registration System API 2.55 API)

| | |
|------------------|---|
| | java.lang.String password) POST method for performing a login, accepting data of type "application/x-www-form-urlencoded". |
| java.lang.String | <u>sanityCheck</u> (java.lang.String onBehalfOf) Sanity check. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

LoginResource

```
public LoginResource()
```

Method Detail

sanityCheck

```
public java.lang.String sanityCheck(@HeaderParam(value="X-On-Behalf-Of")  
                                     java.lang.String onBehalfOf)
```

Sanity check. Used for test purposes only.

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---|
| 200 | OK | always returned with the onBehalfOf parameter in the body |

Parameters:

onBehalfOf - dummy parameter that will be returned as the response

Returns:

the response

postLogin

```
public gov.nasa.eosdis.urs.core.objects.LoginResponse postLogin(@HeaderParam(value="authorization")  
                                                               java.lang.String authorization,  
                                                               @HeaderParam(value="X-On-Behalf-Of")  
                                                               java.lang.String onBehalfOf,  
                                                               LoginRequest request)
```

POST method for performing a user authentication. Authentication operations may include request for user attribute data to be returned upon successful authentication.

Overview (User Registration System API 2.55 API)

Examples:

XML encoded login request

```
curl -d ' \
      <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
      <login> \
        <password>ExamplePassword</password> \
        <userFields/> \
        <username>ExampleUser</username> \
      </login> \
    ' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/xml"
https://<authentication-host>:<port>/login

<ExampleUser>
  - The username for the requested login
<ExamplePassword>
  - The password of the username
<EncodedExampleAuthorization>
  - The basic HTTP authentication header parameter
<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
Response
>a LoginResponse object containing whole user information in XML format.
```

XML encoded login request with <userField>

```
curl -d ' \
      <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
      <login> \
        <password>ExamplePassword</password> \
        <userFields> \
          <userField>firstName</userField> \
          <userField>emailAddress</userField> \
        </userFields> \
        <username>ExampleUser</username> \
      </login> \
    ' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/xml" \
https://<authentication-host>:<port>/login

<ExampleUser>
  - The username for the requested login
<ExamplePassword>
  - The password of the username
<EncodedExampleAuthorization>
  - The basic HTTP authentication header parameter
<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
Response
>a LoginResponse object containing user selected information in XML format.
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Authentication request was accepted and processed. The client must check the response contents to confirm whether or not the authentication was permitted. If not, the response will contain an error message giving the reason for the denial. |
| 400 | Bad Request | The request was incomplete - for example, the user name or user password were not provided. |
| 401 | Unauthorized | The authorization credentials supplied with the request were incomplete or invalid. |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to perform the authentication request, or access user attribute data if requesting fields. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
onBehalfOf - If provided, indicates that the authentication is being performed on behalf of an application (other than that identified by the credentials).
request - a LoginRequest object in the form of an xml string

Returns:

xml string representation of a user if successful. A 404 Error is returned if the user is not found or the authentication fails.

postLoginUrlEncoded

```
public gov.nasa.eosdis.urs.core.objects.LoginResponse postLoginUrlEncoded(@HeaderParam(value="authorization")
    java.lang.String authorization,
    @HeaderParam(value="X-On-Behalf-Of",
    java.lang.String onBehalfOf,
    java.lang.String username,
    java.lang.String password)
```

POST method for performing a login, accepting data of type "application/x-www-form-urlencoded".

Example:

Form encoded login request (most likely from a html form submit)

```
curl -d 'userName=<ExampleUser>&password=<ExamplePassword>' \
--header "authorization:Basic <EncodedExampleAuthorization>" \
https://<authentication-host>:<port>/login

<ExampleUser>
    - The username for the requested login
<ExamplePassword>
    - The password of the username
<EncodedExampleAuthorization>
```

Overview (User Registration System API 2.55 API)

- The basic HTTP authentication header parameter

<authentication-host>

- The host that URS API is running on

<port>

- The port number of the URS API

Response

a LoginResponse object containing whole user information in XML format.

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Authentication request was accepted and processed. The client must check the response contents to confirm whether or not the authentication was permitted. If not, the response will contain an error message giving the reason for the denial. |
| 400 | Bad Request | The request was incomplete - for example, the user name or user password were not provided. |
| 401 | Unauthorized | The authorization credentials supplied with the request were incomplete or invalid. |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to perform the authentication request, or access user attribute data if requesting fields. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication -
http://en.wikipedia.org/wiki/Basic_access_authentication)

onBehalfOf - If provided, indicates that the authentication is being performed on behalf of an application (other than that identified by the credentials).

username - a form parameter (most likely from a html form submit) containing the username for the requested login

password - a form parameter (most likely from a html form submit) containing the password for the requested login

Returns:

xml string representation of a user if successful. A 404 Error is returned if the user is not found or the authentication fails.

gov.nasa.eosdis.urs.api

Class RpcResource

```
java.lang.Object
└─gov.nasa.eosdis.urs.api.URSBaseResource
    └─gov.nasa.eosdis.urs.api.RpcResource
```

```
public class RpcResource
extends URSBaseResource
```

Server side REST interface for performing RPC operations. This interface performs higher level (application type) operations, including requesting a new password, sending out a user name reminder, and activating an account.

Endpoints:

root-url - the path taken to reach the endpoint. For example, it can be thought of as: https://<authentication-host>:<port>/groups or https://<authentication-host>:<port>/groups

```
<root-url>/rpc/version
    See getVersion\(\)
<root-url>/rpc/resetpassword
    See resetPassword\(String, String\)
<root-url>/rpc/retrieveusername
    See retrieveUsername\(String, String\)
<root-url>/rpc/verify
    See verifyAccount\(String\)
<root-url>/rpc/activate
    See activateAccount\(String\)
<root-url>/rpc/appcontent/uid
    See getCustomAppContent\(String, String, String, List\)
<root-url>/rpc/appcontent/uid
    See setCustomAppContent\(String, String, String, String\)
```

Constructor Summary

[RpcResource\(\)](#)

Method Summary

| | |
|---------------------------|--|
| javax.ws.rs.core.Response | <u>activateAccount</u> (java.lang.String activationCode) Account activation method. |
| javax.ws.rs.core.Response | <u>getCustomAppContent</u> (java.lang.String authorization, java.lang.String uid, java.lang.String appId, java.util.List<java.lang.String> keys) The methods is used by applications to retrieve application specific data. |

Overview (User Registration System API 2.55 API)

| | |
|---------------------------|--|
| java.lang.String | <u>getVersion()</u> GET method to get the version of the URS API. |
| void | <u>resetPassword</u> (java.lang.String authorization, java.lang.String email) POST method for performing a password reset for a user. |
| javax.ws.rs.core.Response | <u>retrieveUsername</u> (java.lang.String authorization, java.lang.String email) Username reminder method. |
| javax.ws.rs.core.Response | <u>setCustomAppContent</u> (java.lang.String authorization, java.lang.String keys, java.lang.String uid, java.lang.String appId, java.lang.String json) This method sets the customAppContent attribute of a specified user in LDAP. |
| javax.ws.rs.core.Response | <u>verifyAccount</u> (java.lang.String verificationCode) Account verification method. |

Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait`

Constructor Detail

RpcResource

```
public RpcResource()
```

Method Detail

getVersion

```
public java.lang.String getVersion()
```

GET method to get the version of the URS API. This method does not need any parameters.

Example:

Get URS API version

```
curl https://<authentication-host>:<port>/rpc/version
```

```
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
```

Returns:

the URS API version in a String.

resetPassword

```
public void resetPassword(@HeaderParam(value="authorization")
                           java.lang.String authorization,
                           java.lang.String email)
    throws gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException,
           gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException,
           gov.nasa.eosdis.urs.core.exception.InvalidValueException,
           gov.nasa.eosdis.urs.core.dao.exception.SearchException,
           gov.nasa.eosdis.urs.core.dao.exception.DaoActionException
```

POST method for performing a password reset for a user. This method will generate a new one-time password, update the corresponding user account to a temporary (T) state, and send out an email with the new password to the email address associated with the account. The temporary (T) state into which the account is put will require the user to enter a new permanent password when they next log in.

Example:

Reset password

```
curl -X POST \
      -d 'email=user@example.com' \
      --header 'authorization:Basic <EncodedExampleAuthorization>' \
      --header 'content-type:application/x-www-form-urlencoded' \
      https://<authentication-host>:<port>/rpc/resetpassword

<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 204 | No Content | Returns successfully, blank body |
| 400 | Bad Request | The request was incomplete - for example, no email was provided. |
| 401 | Unauthorized | The authorization credentials supplied with the request were incomplete or invalid. |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to perform the password reset on the requested account. Generally, this operation should be performed by an application, a profile administrator, or an administrator of a group to which the user account belongs. |
| 404 | Not Found | A request was made to reset the password on a user account that does not exist in the LDAP database. Note that this error can result from using an authority that does not have the necessary permissions to see the user account. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Overview (User Registration System API 2.55 API)

Parameters:

authorization - the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
email - the user's email address associated to the user name

Throws:

gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException
gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException
gov.nasa.eosdis.urs.core.exception.InvalidValueException
gov.nasa.eosdis.urs.core.dao.exception.SearchException
gov.nasa.eosdis.urs.core.dao.exception.DaoActionException

retrieveUsername

```
public javax.ws.rs.core.Response retrieveUsername(@HeaderParam(value="authorization")
                                                 java.lang.String authorization,
                                                 java.lang.String email)
```

Username reminder method. This method requests that an email be sent to the address associated with an account containing the user name of that account.

Example:

Username Reminder

```
curl --header 'authorization:Basic <EncodedExampleAuthorization>' \
      https://<authentication-host>:<port>/rpc/retrieveusername?email=user@example.com

<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 204 | NO_CONTENT | Call was successful and an email has been sent. |
| 400 | Bad Request | The request was incomplete - for example, no email address was provided. |
| 401 | Unauthorized | The authorization credentials supplied with the request were incomplete or invalid. |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to perform the password reset on the requested account. Generally, this operation should be performed by an application, a profile administrator, or an administrator of a group to which the user account belongs. |
| 404 | Not Found | A request was made to reset the password on a user account that does not exist in the LDAP database. Note that this error can result from using an authority that does not have the necessary permissions to see the user account. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |

Overview (User Registration System API 2.55 API)

| | | |
|-----|---------------------|---|
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |
|-----|---------------------|---|

Parameters:

authorization - the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
email - the email address to be looked up

Returns:

Status code in the response

Throws:

`java.io.UnsupportedEncodingException`

activateAccount

```
public javax.ws.rs.core.Response activateAccount(java.lang.String activationCode)
```

Account activation method. This method should not be called directly by applications. It should only be invoked as a result of using a link provided by the User Registration System at the time of account creation. These URLs are generally sent out in emails and contain a unique activation code tied to the account.

Example:

Account Activation

```
curl https://<authentication-host>:<port>/rpc/activate?code=<activation_code>

<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<activation_code>
    - Randomly generated code e-mailed to user
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Returns successfully with a simple message indicating account activation. This occurs when the account creation was not invoked by an application, or the invoking application has no redirection url configured in the database. |
| 303 | OK (with redirection) | Indicates that the account has been successfully activated, and the client should be redirected to another URL. This happens when the account creation was invoked by an application, and the application has a 'redirection url' configured in the database. |
| 400 | Bad request | An attempt was made to call this request without an activation code, or with an activation code that could not be interpreted (e.g. someone trying to fake it) |
| 404 | Not Found | This method was invoked using an activation code that was no longer valid. This can happen for example, when the account for which it was created has expired and been deleted. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | | |

Overview (User Registration System API 2.55 API)

| | |
|---------------------|---|
| Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |
|---------------------|---|

Parameters:

activationCode - the activation code provided when the account was created.

Returns:

the response with the status code

verifyAccount

```
public javax.ws.rs.core.Response verifyAccount(java.lang.String verificationCode)
```

Account verification method. This method should not be called directly by applications. It should only be invoked as a result of using a link provided by the User Registration System when an account is verified. Accounts need to be verified after migration into the URS system from an external system or if the email address of the account changes. These URLs are generally sent out in emails and contain a unique verification code tied to the account.

Example:

Account Verification

```
curl https://<authentication-host>:<port>/rpc/verify?code=<verification_code>

<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<verification_code>
    - Randomly generated code e-mailed to user
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|--|
| 200 | OK | Returns successfully with a simple message indicating account verification. This occurs when the account verification was not invoked by an application, or the invoking application has no redirection url configured in the database. |
| 303 | OK (with redirection) | Indicates that the account has been successfully verified, and the client should be redirected to another URL. This happens when the account verification was invoked by an application, and the application has a 'redirection url' configured in the database. |
| 400 | Bad request | An attempt was made to call this request without a verification code, or with a verification code that could not be interpreted (e.g. someone trying to fake it) |
| 404 | Not Found | This method was invoked using a verification code that was no longer valid. This can happen for example, when the account for which it was created has expired and been deleted. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

verificationCode - the verification code provided when the account was modified.

Returns:

the response with the status code

setCustomAppContent

```
public javax.ws.rs.core.Response setCustomAppContent (@HeaderParam(value="authorization")
                                                 java.lang.String authorization,
                                                 @HeaderParam(value="keys")
                                                 java.lang.String keys,
                                                 java.lang.String uid,
                                                 @HeaderParam(value="Client-Id")
                                                 java.lang.String appId,
                                                 java.lang.String json)
                                                 throws gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException,
                                                 gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException,
                                                 gov.nasa.eosdis.urs.core.exception.InvalidValueException,
                                                 gov.nasa.eosdis.urs.core.dao.exception.SearchException,
                                                 gov.nasa.eosdis.urs.core.dao.exception.DaoActionException
```

This method sets the customAppContent attribute of a specified user in LDAP. The content is inserted by applications. Therefore A user can have multiple entries of customAppContent of different applications. However, an application always overwrites its customAppContent if there is one already. An application cannot see the customAppContents of other applications. Note that the header parameter "application-id" should be an existing application user in LDAP.

Example:

Set customAppContent

```
curl -d '{"key1":"value1", "key2":"value2"}' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/json" \
--header "Client-Id:urs-test" \
http://authentication-host:port/urs-api/rpc/appcontent/uid
```

Set new value of a specific key in CustomAppContent

Given a JSON string {"K1": "V1", "K2": {"K11": "V11", "K22": {"K111": "V111"}, "K33": "V33"}, "K4": "V4", "K5": "V5"}, the following curl command updates the value of "V111" to "NEWVALUE".

```
curl -d 'NEWVALUE' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/json" \
--header "Client-Id:urs-test" \
--header "keys:K2,K22,K111" \
http://authentication-host:port/urs-api/rpc/appcontent/uid
```

```
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<application-id>
    - The application ID that tries to talk to the API
<uid>
    - The user to search and return information for.
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 204 | No Content | Creation (or update) operation succeeded |
| 400 | Bad request | An attempt was made to call this request without an activation code, or with an activation code that could not be interpreted (e.g. someone trying to fake it) |
| 401 | Unauthorized | A request with an invalid authority who cannot be identified. |
| 404 | Not Found | This method was invoked using an activation code that was no longer valid. This can happen for example, when the account for which it was created has expired and been deleted. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication -

http://en.wikipedia.org/wiki/Basic_access_authentication)

keys - should be a string of comma separated JSON keys which points to the value that will be updated

uid - the user name

appId - "Client-Id" from the request header

json - the new value of the specified key.

Returns:

the whole JSON string or the JSON string with specified keys from LDAP

Throws:

gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException

gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException

gov.nasa.eosdis.urs.core.exception.InvalidValueException

gov.nasa.eosdis.urs.core.dao.exception.SearchException

gov.nasa.eosdis.urs.core.dao.exception.DaoActionException

getCustomAppContent

```
public javax.ws.rs.core.Response getCustomAppContent (@HeaderParam(value="authorization")
                                                       java.lang.String authorization,
                                                       java.lang.String uid,
                                                       @HeaderParam(value="Client-Id")
                                                       java.lang.String appId,
                                                       java.util.List<java.lang.String> keys)
```

The methods is used by applications to retrieve application specific data. An application authority passes in the encoded authorization string and its "Client-Id" (appId) in the header, specifies user name in the path and keys it tries to retrieve in the query (these query parameters are optional).

URS API will return the application specific data in a json string to the requester.

Overview (User Registration System API 2.55 API)

Example:

Get customAppContent

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
--header "Client-Id:<application-id>" \
'https://<authentication-host>:<port>/rpc/appcontent/<uid>'
```

Get customAppContent with specified keys

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
--header "Client-Id:<application-id>" \
'https://<authentication-host>:<port>/rpc/appcontent/<uid>?key=<SampleKey1>&#038;<SampleKe
```

Get customAppContent with specified keys from a nested JSON string

Given a JSON string {"K1": "V1", "K2": {"K11": "V11", "K22": {"K111": "V111"}, "K33": "V33"}, "K4": "V4", "K5": "V5"}, the following curl command returns {"K2": {"K33": "V33"}, "K5": "V5"}

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
--header "Client-Id:<application-id>" \
'https://<authentication-host>:<port>/rpc/appcontent/<uid>?key=K2,K33&key=K5'

<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<application-id>
    - The application ID that tries to talk to the API
<verification_code>
    - Randomly generated code e-mailed to user
<uid>
    - The user to search and return information for.
<SampleKey1>
    - The specified key.
<SampleKey2>
    - The specified key.
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Returns successfully with a JSON string of this user's customAppContent. |
| 204 | No Content | If there is no customAppContent associated with this application. Or the application tries to look at the customAppContent fields of other applications |
| 400 | Bad request | An attempt was made to call this request without an activation code, or with an activation code that could not be interpreted (e.g. someone trying to fake it) |
| 401 | Unauthorized | A request with an invalid authority who cannot be identified. |
| 404 | Not Found | This method was invoked using an activation code that was no longer valid. This can happen for example, when the account for which it was created has expired and been deleted. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |

Overview (User Registration System API 2.55 API)

| | | |
|-----|------------------------|---|
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |
|-----|------------------------|---|

Parameters:

authorization - the requesting authority (basic HTTP authentication -
http://en.wikipedia.org/wiki/Basic_access_authentication)

uid - the user name

appId - "Client-Id" from the request header

keys - the specified keys

Returns:

the whole JSON string or the JSON string with specified keys from LDAP

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api Class TestResource

```
java.lang.Object
└─gov.nasa.eosdis.urs.api.URSBaseResource
    └─gov.nasa.eosdis.urs.api.TestResource
```

```
public class TestResource
extends URSBaseResource
```

Tools and utilities used only for testing purposes.

This class is not deployed in the operational system.

Endpoints:

root-url - the path taken to reach the endpoint. For example, it can be thought of as:

<https://<authentication-host>:<port>/groups> or <https://<authentication-host>:<port>/groups>

<root-url>/test/users

◊ This end point accepts one kind of test request:

1. Can accept data submitted (XML or JSON form) to test creation of users.

<root-url>/test/mail

◊ This end point accepts two kind of test requests:

1. Can list mail files in the current mail drop box based on search parameters to test search of mail.
2. Can take in an id to test the search and deletion of a mail file that contains the id.

◊ Additional HTML header parameters are

authentication

- (required) the requesting authority (basic HTTP authentication -
http://en.wikipedia.org/wiki/Basic_access_authentication)

X-Require-Activation

- (required) indicate that the account requires an activation email to be generated.

X-On-Behalf-Of

- (Optional) If provided, indicates that the authentication is being performed on behalf of an application (other than that identified by the credentials). Used when an external GUI forwarded a user to the User Registration System GUI in order to create an account. This is used to provide forwarding details in the account activation link.

Author:

psmith

Constructor Summary

| |
|--|
| <u>TestResource ()</u> |
|--|

Method Summary

| | |
|---------------------------|---|
| javax.ws.rs.core.Response | <u>deleteMail</u> (java.lang.String id) DELETE method for performing the deletion of a mail file in the mail drop box. |
| javax.ws.rs.core.Response | <u>getMail</u> (java.lang.String to, java.lang.String from, java.lang.String subject) GET method which returns all the mail files in the current inbox, with an optional query parameters. |
| javax.ws.rs.core.Response | <u>mockCreateUser</u> (java.lang.String authorization, java.lang.String onBehalfOf, boolean requireActivation, boolean success, gov.nasa.eosdis.urs.core.objects.User user) POST method used to mock user creation operation. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

TestResource

```
public TestResource()
```

Method Detail

mockCreateUser

```
public javax.ws.rs.core.Response mockCreateUser(@HeaderParam(value="authorization")  
                                                java.lang.String authorization,  
                                                @HeaderParam(value="X-On-Behalf-Of")  
                                                java.lang.String onBehalfOf,  
                                                @HeaderParam(value="X-Require-Activation")  
                                                boolean requireActivation,  
                                                boolean success,  
                                                gov.nasa.eosdis.urs.core.objects.User user)
```

POST method used to mock user creation operation.

When LDAP returns 80 error, retries should be made. The number of retries is read from the properties file.

Examples:

XML encoded mock user creation test

```
curl -d ' \  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> \  
<user> \  
    <userName>ExampleUser</userName> \  
    <emailAddress>ExampleEmailAddress</emailAddress> \  
'
```

Overview (User Registration System API 2.55 API)

```

<password>ExamplePassword</password> \
<firstName>ExampleFirstName</firstName> \
<lastName>ExampleLastName</lastName> \
<affiliation>ExampleAffiliation</affiliation> \
<country>ExampleCountry</country> \
</user> \
' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/xml" https://<authentication-host>:<port>/test/users

<ExampleUser>
    - The username for the requested login
<ExampleEmailAddress>
    - The email address of the username
<ExamplePassword>
    - The password of the username
<ExampleFirstName>
    - The first name of the username
<ExampleLastName>
    - The last name of the username
<ExampleAffiliation>
    - The affiliation of the username
<ExampleCountry>
    - The country of the username
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API

```

Response

a User object containing whole user information in XML format.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|--|
| 201 | Created | Creation Succeeded, location to new resource is set in the location field in the header |
| 303 | See Other | Creation Succeeded, occurs when "X-On-Behalf-Of" header value is set to an URS application with a valid ursAppLoginUrl attribute in LDAP. The redirect target is set to the LDAP attribute value |
| 400 | Bad Request | Validation Error @see gov.nasa.eosdis.urs.util.UserValidation for up to date explanation of a valid user |
| | | Portal set to anything other than "FALSE" |
| | | Any value is assigned to adminGroups (not allowed in creation) |
| 401 | Unauthorized | A valid API username and password is required for this operation |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to create the requested account. |
| 409 | Conflict | Non-unique username or email address |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Overview (User Registration System API 2.55 API)

Parameters:

authorization - the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
onBehalfOf - if provided, the uid of an application that initially requested the creation of the user. This parameter is used when an external GUI forwarded a user to the User Registration System GUI in order to create an account. This is used to provide forwarding details in the account activation link.
user - the details of the user account to create
requireActivation - used to indicate that the account requires an activation email to be generated. This defaults to true, and should only be overridden by applications that provide an alternative account verification mechanism, or when migrating existing user accounts from another system.

Returns:

created status code or exceptions.

See Also:

[UserValidation](#)

getMail

```
public javax.ws.rs.core.Response getMail(java.lang.String to,  
                                java.lang.String from,  
                                java.lang.String subject)
```

GET method which returns all the mail files in the current inbox, with an optional query parameters.

This method is used in API integration tests to retrieve generated emails for verification purposes.

Examples:

Search for mail file with one parameter provided: <to> .

```
curl https://<authentication-host>:<port>/test/mail?to=<ExampleTo>
```

Search for mail file with multiple parameters provided: <to> and <subject> .

```
curl https://<authentication-host>:<port>/test/mail?to=<ExampleTo>&subject=<ExampleSubject>
```

<ExampleTo>

- If provided, returns only those emails whose 'to' address matches this value.

<ExampleFrom>

- If provided, returns only those emails whose 'from' address matches this value.

<ExampleSubject>

- If provided, returns only those emails whose 'subject' address matches this value.

<authentication-host>

- The host that URS API is running on

<port>

- The port number of the URS API

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|--|
| 200 | OK (No content) | Retrieval of generated emails for verification has succeeded. |
| 404 | Not Found | Returned if the mail drop box is empty/not configured due to previously not existing. The mail drop box did not exist but was created. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. The configured mail directory does not exist and cannot be created. |

Parameters:

- to - if provided, returns only those emails whose 'to' address matches this value.
- from - if provided, returns only those emails whose 'from' address matches this value.
- subject - if provided, returns only those emails whose 'subject' address matches this value.

Returns:

an xml formatted mail if there is one.

deleteMail

```
public javax.ws.rs.core.Response deleteMail(java.lang.String id)
```

DELETE method for performing the deletion of a mail file in the mail drop box.

This method is used in API integration tests to remove generated emails. It is not an error to request deletion of a mail file that does not exist.

Example:

Mail file search and deletion test.

```
curl -X DELETE https://<authentication-host>:<port>/test/mail/<id>

<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
<id>
    - The id of the email message to delete.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK (No content) | Deletion of mail file in the mail drop box succeeded. |
| 404 | Not Found | Returned if the mail drop box is empty/not configured due to previously not existing. The mail drop box did not exist but was created. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. This error results in the inability to delete the mail file. |

Parameters:

`id` - the id of the email message to delete. Ids are obtained from the email listings returned by the GET operation.

Returns:

a response that indicates whether the deletion is successful.

gov.nasa.eosdis.urs.api Class URSBaseResource

```
java.lang.Object
└ gov.nasa.eosdis.urs.api.URSBaseResource
```

Direct Known Subclasses:

[GroupResource](#), [LoginResource](#), [MetricResource](#), [RpcResource](#), [TestResource](#), [UserResource](#)

```
public class URSBaseResource
extends java.lang.Object
```

The base class URSBaseResource is the super class of classes GroupResource, LoginResource, RpcResource, TestResource and UserResource.

URSBaseResource provides server side methods for ChangeAuthority and validating ChangeAuthority. This class also has methods to log messages, build list of changes made to a user, and to determine if a password and user account are being changed.

Constructor Summary

[URSBaseResource\(\)](#)

Method Summary

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

URSBaseResource

```
public URSBaseResource()
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api

Class UserResource

```
java.lang.Object
└─gov.nasa.eosdis.urs.api.URSBaseResource
    └─gov.nasa.eosdis.urs.api.UserResource
```

```
public class UserResource
extends URSBaseResource
```

Server side REST interface for performing user based operations, such as create, read, update, and delete.

Endpoints:

root-url - the path taken to reach the endpoint. For example, it can be thought of as:

<https://<authentication-host>:<port>/groups> or <https://<authentication-host>:<port>/groups>

<root-url>/users

- ◊ Handles operations for users such as create, update, delete and read/search.
- ◊ This end point accepts five kinds of requests:
 1. It accepts data submitted in JSON or XML form to create a user.
 2. Given a user name, it can request for that user to be deleted in the database.
 3. Can retrieve either all or specific information on a certain user.
 4. It can search for information about users found in the database that have matching values to the parameters given.
 5. Handles a passed in XML object to update a user.
- ◊ Additional HTML header parameters are
 - authentication
 - (required) the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
 - X-Require-Activation
 - (required) indicate that the account requires an activation email to be generated.
 - X-On-Behalf-Of
 - (Optional) If provided, indicates that the authentication is being performed on behalf of an application (other than that identified by the credentials).
 - X-Send-Mail
 - (Optional) If provided, indicates whether an email should be sent informing users of the update.
 - X-Update-Reason
 - (Optional) If provided, indicates reason for the update.

Constructor Summary

| |
|--|
| <u>UserResource ()</u> |
|--|

Method Summary

| | |
|---------------------------------------|--|
| javax.ws.rs.core.Response | <code>createUser</code> (java.lang.String authorization, java.lang.String onBehalfOf, boolean requireActivation, gov.nasa.eosdis.urs.core.objects.User user) User 'Create' operation. |
| void | <code>deleteUser</code> (java.lang.String authorization, java.lang.String uid) User 'delete' operation. |
| gov.nasa.eosdis.urs.core.objects.User | <code>getUser</code> (java.lang.String authorization, java.lang.String uid, java.util.List<java.lang.String> fields) User 'Get' operation. |
| javax.ws.rs.core.Response | <code>getUsers</code> (java.lang.String authorization, java.lang.String userName, java.lang.String email, java.lang.String name, java.lang.String lastName, java.lang.String firstName, java.lang.String memGroup, java.lang.String adminGroup, java.lang.String status, java.lang.String organization, java.lang.String sortBy, int pageSize, int offset, boolean rtnPassword, boolean wildcardSearch, java.lang.String order) Search users by a variety of fields. |
| javax.ws.rs.core.Response | <code>updateUser</code> (java.lang.String authorization, java.lang.String updateReason, boolean sendMail, java.lang.String uid, gov.nasa.eosdis.urs.core.objects.User updatedUser) User 'Update' operation. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

UserResource

```
public UserResource()
```

Method Detail

getUsers

```
public javax.ws.rs.core.Response getUsers (@HeaderParam(value="authorization")
                                              java.lang.String authorization,
                                              java.lang.String userName,
                                              java.lang.String email,
                                              java.lang.String name,
                                              java.lang.String lastName,
                                              java.lang.String firstName,
                                              java.lang.String memGroup,
                                              java.lang.String adminGroup,
                                              java.lang.String status,
                                              java.lang.String organization,
                                              java.lang.String sortBy,
                                              int pageSize,
                                              int offset,
                                              boolean rtnPassword,
                                              boolean wildcardSearch,
                                              java.lang.String order)
```

Search users by a variety of fields.

Note that search access to the database is restricted by hiding entries from unprivileged users. This means that different users may receive different results when submitting the same query.

Example:

Search user request

```
curl 'https://<authentication-host>:<port>/users/?<searchBy>=<value>'
      --header "authorization:Basic <EncodedExampleAuthorization>" \
```

Search user request by username

```
curl 'https://<authentication-host>:<port>/users/?username=<ExampleUserName>'
      --header "authorization:Basic <EncodedExampleAuthorization>" \
```

Search user request by memGroup and sort by username

```
curl 'https://<authentication-host>:<port>/users/?status=<ExampleStatus>&sortBy=username'
      --header "authorization:Basic <EncodedExampleAuthorization>" \
```

The following example finds users with email address being exactly 'test_*@example.com', since it is not a wildcard search (wildcard=false). i.e. it will not match emails like 'test_1@example.com' or 'test_abc@example.com'.

```
curl 'https://<authentication-host>:<port>/users/?email=test_*@example.com&wildcard=false'
      --header "authorization:Basic <EncodedExampleAuthorization>" \
```

The following example sort the results based on the given 'order' parameter which accepts either asc or desc. If 'sortBy' is specified but 'order' is not, the returned results default to ascend order.

```
curl 'https://<authentication-host>:<port>/users/?username=<ExampleUserName>&sortBy=ursFirstName'
      --header "authorization:Basic <EncodedExampleAuthorization>" \
```

<authentication-host>
- The host that URS API is running on

<port>
- The port number of the URS API

<value>
- The value to be used in your search.

<searchBy>
- The value field to search by. Can be "username", "email", "name", "lastName", "firstName", "memGroup", "adminGroup", "status", and "organization". Can also format search by providing along

Overview (User Registration System API 2.55 API)

with the listed values options for ways to sort, number of results to retrieve, return password, and offset value. All by the options "sortBy", "pageSize", "rtnPassword" and "offset".

<EncodedExampleAuthorization>

- The basic HTTP authentication header parameter.

<ExampleUserName>

- Example username to use in search .

<ExampleStatus>

- Example status to use in search .

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Search succeeded with results |
| 400 | Bad Request | This error is returned |
| 401 | Unauthorized | The authorization credentials supplied with the request were incomplete or invalid. |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to perform the password reset on the requested account. Generally, this operation should be performed by an application, a profile administrator, or an administrator of a group to which the user account belongs. |
| 404 | Not Found | No search results were returned. NB. This is still TBD. An empty results set may be returned with a 200 (OK) status. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication -

http://en.wikipedia.org/wiki/Basic_access_authentication)

userName - if provided, search for users with the given userName. May include '*' wildcard.

email - if provided, search for users with the given email address. May include '*' wildcard.

name - if provided, searches against both the first name and the last name (or search). May include '*' wildcards.

lastName - if provided, searches against the last name. May include '*' wildcards.

firstName - if provided, searches against the first name. May include '*' wildcards.

memGroup - if provided, searches against a user's group membership list.

adminGroup - if provided, searches against a user's group administration list.

status - if provided, searches against the current account status.

organization - if provided, searches against a user's organization field. May include '*' wildcards.

sortBy - specifies the field by which the results should be sorted. If not provided, results will be sorted alphabetically by username (ascending).

pageSize - the number of results to retrieve. Query results are returned in pages. The default page size is 20.

offset - the index of the first result of the page being returned. This value can be used to page through results. It defaults to 0.

rtnPassword - if set to true, the search results will contain passwords. Otherwise, they will be set to null.

wildcardSearch - indicates whether the search should escape wildcards. false = escape

order - indicates the order of the returned search results. (ASC/DESC)

Returns:

a list of users satisfying the search

createUser

```
public javax.ws.rs.core.Response createUser(@HeaderParam(value="authorization")
                                              java.lang.String authorization,
                                              @HeaderParam(value="X-On-Behalf-Of")
                                              java.lang.String onBehalfOf,
                                              @HeaderParam(value="X-Require-Activation")
                                              boolean requireActivation,
                                              gov.nasa.eosdis.urs.core.objects.User user)
```

User 'Create' operation.

This POST operation is used to create new user accounts in URS. Note that user create permissions are generally restricted to applications, group administrators, and profile administrators.

Examples:

XML encoded user creation test

```
curl -d ' \
      <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
      <user> \
        <userName>ExampleUser</userName> \
        <emailAddress>ExampleEmailAddress</emailAddress> \
        <password>ExamplePassword</password> \
        <firstName>ExampleFirstName</firstName> \
        <lastName>ExampleLastName</lastName> \
        <affiliation>ExampleAffiliation</affiliation> \
        <country>ExampleCountry</country> \
      </user> \
    ' \
    --header "authorization:Basic EncodedExampleAuthorization" \
    --header "X-Require-Activation:true" \
    --header "content-type:application/xml" \
    https://<authentication-host>:<port>/users

<ExampleUser>
  - The username for the requested login
<ExampleEmailAddress>
  - The email address of the username
<ExamplePassword>
  - The password of the username
<ExampleFirstName>
  - The first name of the username
<ExampleLastName>
  - The last name of the username
<ExampleAffiliation>
  - The affiliation of the username
<ExampleCountry>
  - The country of the username
<EncodedExampleAuthorization>
  - The basic HTTP authentication header parameter
<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
Response
```

Overview (User Registration System API 2.55 API)

a User object containing whole user information in XML format.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|--|
| 201 | Created | Creation Succeeded, location to new resource is set in the location field in the header |
| 303 | See Other | Creation Succeeded, occurs when "X-On-Behalf-Of" header value is set to an URS application with a valid ursAppLoginUrl attribute in LDAP. The redirect target is set to the LDAP attribute value |
| 400 | Bad Request | Validation Error @see gov.nasa.eosdis.urs.util.UserValidation for up to date explanation of a valid user |
| | | Portal set to anything other than "FALSE" |
| | | Any value is assigned to adminGroups (not allowed in creation) |
| 401 | Unauthorized | A valid API username and password is required for this operation |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to create the requested account. |
| 409 | Conflict | Non-unique username or email address |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication -

http://en.wikipedia.org/wiki/Basic_access_authentication)

onBehalfOf - if provided, the uid of an application that initially requested the creation of the user. This parameter is used when an external GUI forwarded a user to the User Registration System GUI in order to create an account. This is used to provide forwarding details in the account activation link.

requireActivation - used to indicate that the account requires an activation email to be generated. This defaults to true, and should only be overridden by applications that provide an alternative account verification mechanism, or when migrating existing user accounts from another system.

user - the details of the user account to create

Returns:

the username of the newly created user

See Also:

[UserValidation](#)

getUser

```
public gov.nasa.eosdis.urs.core.objects.User getUser(@HeaderParam(value="authorization")
                                                 java.lang.String authorization,
                                                 java.lang.String uid,
                                                 java.util.List<java.lang.String> fields)
```

User 'Get' operation.

This method is used to retrieve user account details. Permission controls are applied in both the API and the database, and will prohibit a client with insufficient privileges from retrieving user data.

Example:

Overview (User Registration System API 2.55 API)

Get User Request

```
curl 'https://<authentication-host>:<port>/users/<uid>?field=<field>' \
--header "authorization:Basic <EncodedExampleAuthorization>" \
```

Get user request to return all information of user.

```
curl 'https://<authentication-host>:<port>/users/<uid>?field=all' \
--header "authorization:Basic <EncodedExampleAuthorization>" \
```

Get user request to return only specified information of user, for example: email address.

```
curl 'https://<authentication-host>:<port>/users/<uid>?field=emailAddress' \
--header "authorization:Basic <EncodedExampleAuthorization>" \
```

<authentication-host>
- The host that URS API is running on

<port>
- The port number of the URS API

<uid>
- The user to search and return information for.

<field>
- The user field to search for. If all, will return all information.

<EncodedExampleAuthorization>
- The basic HTTP authentication header parameter.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | user is found and returned (either with full or partial attributes, as per the request) |
| 400 | Bad Request | Returned if an invalid user name was provided (should not be confused with a missing user; represents an invalid username) |
| | | Returned if the request included non-valid user attributes (for partial attribute requests) |
| 404 | Not Found | Returned if the user is not found in the User Registration System. Note that this can result from a lack of permissions, since user accounts are hidden to unauthorized clients |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication -

http://en.wikipedia.org/wiki/Basic_access_authentication)

uid - the user name (uid) of the user to be acted on

fields - specify what fields are request to return. If "all" is specified, all other fields will be ignored.

Returns:

the user object associated with the user name

updateUser

```
public javax.ws.rs.core.Response updateUser(@HeaderParam(value="authorization")
                                             java.lang.String authorization,
                                             @HeaderParam(value="X-Update-Reason")
                                             java.lang.String updateReason,
                                             @HeaderParam(value="X-Send-Mail")
                                             boolean sendMail,
```

Overview (User Registration System API 2.55 API)

```
java.lang.String uid,  
gov.nasa.eosdis.urs.core.objects.User updatedUser)
```

User 'Update' operation.

NOTE: 3/28/2012 (KEB) To deal with migration issues and issues requiring that any email changes to an account be verified via the RpcResource, if the account verified date is included in the update, but the email address is also changed OR the existing account verified date is null, the updated value is IGNORED. We will log the event, but not update LDAP with the supplied value. The logic for this is handled in the URSBaseResource class.

Example:

Update Request

```
curl -X PUT -d '  
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \  
        <user> \  
            <firstName>UpdatedFirstName</firstName> \  
            <lastName>UpdatedLastName</lastName> \  
            <emailAddress>UpdatedEmailAddress</emailAddress> \  
            <affiliation>UpdatedAffiliation</affiliation> \  
            <country>UpdatedCountry</country> \  
        </user> \  
    ' \  
    --header "authorization:Basic <EncodedExampleAuthorization>" \  
    --header "X-Send-Mail:true" \  
    --header "Accept: application/xml" \  
'https://<authentication-host>:<port>/users/<uid>'  
  
<authentication-host>  
    - The host that URS API is running on  
<port>  
    - The port number of the URS API  
<uid>  
    - The user to update.  
<UpdatedFirstName>  
    - Update value for uid first name.  
<UpdatedLastName>  
    - Update value for uid last name.  
<UpdatedEmailAddress>  
    - Update value for uid email address.  
<UpdatedAffiliation>  
    - Update value uid for affiliation.  
<UpdatedCountry>  
    - Update value for uid country.  
<EncodedExampleAuthorization>  
    - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 204 | OK (No content) | Update to user succeeded |
| 400 | Bad Request | User attribute validation error Missing update reason in the header "X-Update-Reason" (if called by an authorizing user that is an admin) |

Overview (User Registration System API 2.55 API)

| | | |
|-----|-----------------------|---|
| | | If the new password has changed but is encrypted, or if it fails password validation. |
| 401 | Unauthorized | A valid API username and password is required for this operation or authorizing user is not found |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to update the requested account fields. Certain fields within a user account are restricted, and may not be updated, or updated only by privileged clients. |
| 404 | NotFound | The user to be updated could not be found. |
| 409 | Conflict | Non-unique email address on the update |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication - http://en.wikipedia.org/wiki/Basic_access_authentication)
 updateReason - (optional) the reason for the update
 sendMail - flag to indicate whether an email should be sent informing users of the update.
 uid - the user name (uid) of the user to be acted on
 updatedUser - the updated user object

Returns:

result of the update operation

See Also:

[UserValidation](#)

deleteUser

```
public void deleteUser(@HeaderParam(value="authorization")
                      java.lang.String authorization,
                      java.lang.String uid)
```

User 'delete' operation.

Example:

Delete Request

```
curl -X DELETE --header "authorization:Basic <EncodedExampleAuthorization>" \
      'https://<authentication-host>:<port>/users/<uid>'

<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
<uid>
  - The user to delete.
<EncodedExampleAuthorization>
  - The basic HTTP authentication header parameter.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---------------------------|
| | | |

Overview (User Registration System API 2.55 API)

| | | |
|-----|-----------------------|--|
| 204 | OK (No content) | Deletion of user succeeded |
| 401 | Unauthorized | A valid API username and password is required for this operation |
| 403 | Forbidden | The authorization credentials supplied with the request were insufficient to delete the user account. Account deletion privileges are generally restricted to user profile administrators. |
| 404 | NotFound | The user to be updated could not be found. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |
| 503 | Service Unavailable | This error should be very infrequent, but usually indicates a temporary problem, such as an internal communication error or temporary loss of access to the database. |

Parameters:

authorization - the requesting authority (basic HTTP authentication -
http://en.wikipedia.org/wiki/Basic_access_authentication)
uid - the user name (uid) of the user to be acted on

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api

Class UserValidationResource

```
java.lang.Object
└─ gov.nasa.eosdis.urs.api.UserValidationResource
```

```
public class UserValidationResource
extends java.lang.Object
```

Server-side REST interface for validating various components of a user before creation of or updates to that user.

Note that this unlike @see ValidResource, this class *does* perform validation. Any errors in validating the supplied data are returned through the [ValidationError](#) and [ValidationErrors](#) classes.

Endpoints:

<root-url>/valids/user/[action-to-take]

- ◊ Additional HTML header parameters are authentication
 - (required) the requesting authority ([basic HTTP authentication](#))

Constructor Summary

| | |
|--|--|
| UserValidationResource() | |
|--|--|

Method Summary

| | |
|----------------------------------|--|
| ValidationErrors | validateEmailFromQuery (java.lang.String email) POST method for validating a user's email address. |
| ValidationErrors | validateEmailFromUser (java.lang.String requestBody, java.lang.String contentType) POST method for validating a submitted user's email address (usually for updating). |
| ValidationErrors | validatePasswordFromQuery (java.lang.String password) POST method for validating a user's password. |
| ValidationErrors | validatePasswordFromUser (java.lang.String requestBody, java.lang.String contentType) POST method for validating a submitted user's password (usually for updating). |
| ValidationErrors | validateUpdatedUser (java.lang.String requestBody, java.lang.String contentType) POST method for validating a submitted user resource (usually for updating). |
| ValidationErrors | validateUser (java.lang.String requestBody, java.lang.String contentType) |

Overview (User Registration System API 2.55 API)

| | |
|-------------------------|---|
| | POST method for validating a submitted user resource. |
| <u>ValidationErrors</u> | validateUserAlias (java.lang.String requestBody, java.lang.String contentType) POST method for validating a submitted user resource. |
| <u>ValidationErrors</u> | validateUsernameFromQuery (java.lang.String uid) POST method for validating a user's username. |
| <u>ValidationErrors</u> | validateUsernameFromUser (java.lang.String requestBody, java.lang.String contentType) POST method for validating a submitted user's username (usually for updating). |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

UserValidationResource

```
public UserValidationResource()
```

Method Detail

validateUser

```
public ValidationErrors validateUser(java.lang.String requestBody,  
@HeaderParam(value="content-type")  
java.lang.String contentType)
```

POST method for validating a submitted user resource.

This method should be used to check all fields for validity for a new user to the system.

An alias exists for this request: */validation/user/create* ([validateUserAlias\(String, String\)](#)). That aliased path performs the same functionality as this method.

To update a user, use the */validation/user/update* path ([validateUpdatedUser\(String, String\)](#)).

This method can handle both XML and JSON representations of a User.

Examples:

XML encoded user validation request

```
curl -d ' \  
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \  
    <user> \  
        <userFields /> \  
'
```

Overview (User Registration System API 2.55 API)

```
<username>ExampleUser</username> \
</user> \
' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:application/xml" \
https://<authentication-host>:<port>/validation/user

<ExampleUser>
    - The username for the user to be validated
<userFields>
    - Fields to be validated, such as emailAddress and firstName
<EncodedExampleAuthorization>
    - The basic HTTP authentication header parameter
<authentication-host>
    - The host that URS API is running on
<port>
    - The port number of the URS API
Response
    Errors that occurred in validation (via XML representation of ValidationErrors) or an empty response on success.
```

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response. |
| 204 | No Content | Validation was performed and no errors were found. |

Parameters:

requestBody - the User to validate in the form of an XML fragment or JSON stringified object.
contentType - a String indicating the incoming format of the User object. If this is not MediaType.APPLICATION_JSON, then the User is assumed to be XML and parsed accordingly.

validateUserAlias

```
public ValidationErrors validateUserAlias(java.lang.String requestBody,
                                         @HeaderParam(value="content-type")
                                         java.lang.String contentType)
```

POST method for validating a submitted user resource.

This is an alias to validateUser(String, String), and operates according to that method's documentation.

Status Code:

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response. |
| 204 | No Content | Validation was performed and no errors were found. |

Overview (User Registration System API 2.55 API)

Parameters:

requestBody - the User to validate in the form of an XML fragment or JSON stringified object.
contentType - a String indicating the incoming format of the User object. If this is not
MediaType.APPLICATION_JSON, then the User is assumed to be XML and parsed accordingly.

Returns:

null if there are no validation errors; otherwise, a [ValidationErrors](#) object containing a list of [ValidationErrors](#) representing problems that occurred during validation of the supplied data.

validateUpdatedUser

```
public ValidationErrors validateUpdatedUser(java.lang.String requestBody,  
                                         @HeaderParam(value="content-type")  
                                         java.lang.String contentType)
```

POST method for validating a submitted user resource (usually for updating).

This method should be used to check only a subset of a user's fields for validity, such as when updating a user.

To create a user and validate all fields, use the */validation/user* path ([validateUser\(String, String\)](#)).

This method can handle both XML and JSON representations of a User.

Examples:

XML encoded user validation request

```
curl -d ' \  
         <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \  
         <user> \  
             <userFields /> \  
             <username>ExampleUser</username> \  
         </user> \  
     ' \  
     --header "authorization:Basic EncodedExampleAuthorization" \  
     --header "content-type:application/xml" \  
     https://<authentication-host>:<port>/validation/user/update
```

<ExampleUser>

- The username for the user to be validated

<userFields>

- Fields to be validated, such as emailAddress and firstName

<EncodedExampleAuthorization>

- The basic HTTP authentication header parameter

<authentication-host>

- The host that URS API is running on

<port>

- The port number of the URS API

Response

Errors that occurred in validation (via XML representation of [ValidationErrors](#)) or an empty response on success.

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the |

Overview (User Registration System API 2.55 API)

| response | | |
|----------|------------|---|
| 204 | No Content | Validation was performed and no errors were found |

Parameters:

requestBody - the User to validate in the form of an XML fragment or JSON stringified object.
contentType - a String indicating the incoming format of the User object. If this is not
MediaType.APPLICATION_JSON, then the User is assumed to be XML and parsed accordingly.

Returns:

null if there are no validation errors; otherwise, a `ValidationErrors` object containing a list of
`ValidationErrors` representing problems that occurred during validation of the supplied data.

validatePasswordFromQuery

```
public ValidationErrors validatePasswordFromQuery(java.lang.String password)
```

POST method for validating a user's password.

This method should be used to check only a specific field; other fields (if the request contains an XML- or JSON-encoded user object) are ignored in validation.

To create a user and validate all fields, use the `/validation/user` path (`validateUser(String, String)`).

This method is most commonly used with a form-encoded parameter, like
`password=[password_to_validate]`.

Examples:

Form-encoded user validation request

```
curl -d 'password=[password_to_validate]' \
      --header "authorization:Basic EncodedExampleAuthorization" \
      --header "content-type:application/x-www-form-urlencoded" \
      https://<authentication-host>:<port>/validation/user/password
```

`password_to_validate`
- The potential password as a string
<EncodedExampleAuthorization>
- The basic HTTP authentication header parameter
<authentication-host>
- The host that URS API is running on
<port>
- The port number of the URS API

Response

Errors that occurred in validation (via XML representation of `ValidationErrors`) or an empty response on success.

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response |
| 204 | No Content | Validation was performed and no errors were found |

Parameters:

`password` - the String to validate as a password

Returns:

`null` if there are no validation errors; otherwise, a `ValidationErrors` object containing a list of `ValidationErrors` representing problems that occurred during validation of the supplied data.

validatePasswordFromUser

```
public ValidationErrors validatePasswordFromUser(java.lang.String requestBody,
                                              @HeaderParam(value="content-type")
                                              java.lang.String contentType)
```

POST method for validating a submitted user's password (usually for updating).

This method should be used to check only a user's password.

To create a user and validate all fields, use the `/validation/user` path (`validateUser(String, String)`).

This method can handle both XML and JSON representations of a User.

Examples:

XML encoded user validation request

```
curl -d ' \
      <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \
      <user> \
        <userFields /> \
        <username>ExampleUser</username> \
        <password>password_to_validate</password> \
      </user> \
    ' \
    --header "authorization:Basic EncodedExampleAuthorization" \
    --header "content-type:application/xml" \
    https://<authentication-host>:<port>/validation/user/password

<ExampleUser>
  - The username for the user to be validated
<userFields>
  - User fields not validated, such as accountVerifiedDate and firstName
<password_to_validate>
  - The password to validate for this user.
<EncodedExampleAuthorization>
  - The basic HTTP authentication header parameter
<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
Response
  Errors that occurred in validation (via XML representation of ValidationErrors) or an empty response on success.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response |

Overview (User Registration System API 2.55 API)

| | | |
|-----|------------|---|
| 204 | No Content | Validation was performed and no errors were found |
|-----|------------|---|

Parameters:

requestBody - the User to validate in the form of an XML fragment or JSON stringified object.
contentType - a String indicating the incoming format of the User object. If this is not
MediaType.APPLICATION_JSON, then the User is assumed to be XML and parsed accordingly.

Returns:

null if there are no validation errors; otherwise, a `ValidationErrors` object containing a list of
`ValidationErrors` representing problems that occurred during validation of the supplied data.

validateUsernameFromUser

```
public ValidationErrors validateUsernameFromUser(java.lang.String requestBody,  
                                                 @HeaderParam(value="content-type")  
                                                 java.lang.String contentType)
```

POST method for validating a submitted user's username (usually for updating).

This method should be used to check only a user's username.

To create a user and validate all fields, use the `/validation/user` path (`validateUser(String, String)`).

This method can handle both XML and JSON representations of a User.

Examples:

XML encoded user validation request

```
curl -d ' \  
         <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \  
         <user> \  
             <userFields /> \  
             <username>ExampleUser</username> \  
         </user> \  
     ' \  
     --header "authorization:Basic EncodedExampleAuthorization" \  
     --header "content-type:application/xml" \  
     https://<authentication-host>:<port>/validation/user/username
```

```
<ExampleUser>  
    - The username for the user that should be validated  
<userFields>  
    - User fields not validated, such as accountVerifiedDate and firstName  
<EncodedExampleAuthorization>  
    - The basic HTTP authentication header parameter  
<authentication-host>  
    - The host that URS API is running on  
<port>  
    - The port number of the URS API
```

Response

Errors that occurred in validation (via XML representation of `ValidationErrors`) or an empty response on success.

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|---------------------------|
|-------------|------------------|---------------------------|

Overview (User Registration System API 2.55 API)

| | | |
|-----|------------|--|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response |
| 204 | No Content | Validation was performed and no errors were found |

Parameters:

requestBody - the User to validate in the form of an XML fragment or JSON stringified object.
 contentType - a String indicating the incoming format of the User object. If this is not
 MediaType.APPLICATION_JSON, then the User is assumed to be XML and parsed accordingly.

Returns:

null if there are no validation errors; otherwise, a ValidationErrors object containing a list of ValidationErrors representing problems that occurred during validation of the supplied data.

validateUsernameFromQuery

```
public ValidationErrors validateUsernameFromQuery(java.lang.String uid)
```

POST method for validating a user's username.

This method should be used to check only a specific field; other fields (if the request contains an XML- or JSON-encoded user object) are ignored in validation.

To create a user and validate all fields, use the */validation/user* path (validateUser(String, String)).

This method is most commonly used with a form-encoded parameter, like
 username=[username_to_validate].

Examples:

Form-encoded user validation request

```
curl -d 'username=[username_to_validate]' \
--header "authorization:Basic EncodedExampleAuthorization" \
--header "content-type:x-www-form-urlencoded" \
https://<authentication-host>:<port>/validation/user/password
```

```
username_to_validate
- The potential username as a string
<EncodedExampleAuthorization>
- The basic HTTP authentication header parameter
<authentication-host>
- The host that URS API is running on
<port>
- The port number of the URS API
```

Response

Errors that occurred in validation (via XML representation of ValidationErrors) or an empty response on success.

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response |
| 204 | No Content | Validation was performed and no errors were found |

Parameters:

Overview (User Registration System API 2.55 API)

uid - the String to validate as a username

Returns:

null if there are no validation errors; otherwise, a `ValidationErrors` object containing a list of `ValidationErrors` representing problems that occurred during validation of the supplied data.

validateEmailFromUser

```
public ValidationErrors validateEmailFromUser(java.lang.String requestBody,  
                                              @HeaderParam(value="content-type")  
                                              java.lang.String contentType)
```

POST method for validating a submitted user's email address (usually for updating).

This method should be used to check only a user's email address.

To create a user and validate all fields, use the `/validation/user` path (`validateUser(String, String)`).

This method can handle both XML and JSON representations of a User.

Examples:

XML encoded user validation request

```
curl -d ' \  
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?> \  
    <user> \  
        <userFields /> \  
        <username>ExampleUser</username> \  
        <emailaddress>emailaddress_to_validate</emailaddress> \  
    </user> \  
    ' \  
    --header "authorization:Basic EncodedExampleAuthorization" \  
    --header "content-type:application/xml" \  
    https://<authentication-host>:<port>/validation/user/password  
  
<ExampleUser>  
    - The username for the user to be validated  
<userFields>  
    - User fields not validated, such as accountVerifiedDate and firstName  
<emailaddress_to_validate>  
    - The email address to validate for this user.  
<EncodedExampleAuthorization>  
    - The basic HTTP authentication header parameter  
<authentication-host>  
    - The host that URS API is running on  
<port>  
    - The port number of the URS API  
Response  
    Errors that occurred in validation (via XML representation of ValidationErrors) or an empty response on success.
```

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response |

Overview (User Registration System API 2.55 API)

| | | |
|-----|------------|---|
| 204 | No Content | Validation was performed and no errors were found |
|-----|------------|---|

Parameters:

requestBody - the User to validate in the form of an XML fragment or JSON stringified object.
contentType - a String indicating the incoming format of the User object. If this is not
MediaType.APPLICATION_JSON, then the User is assumed to be XML and parsed accordingly.

Returns:

null if there are no validation errors; otherwise, a [ValidationErrors](#) object containing a list of
[ValidationErrors](#) representing problems that occurred during validation of the supplied data.

validateEmailFromQuery

```
public ValidationErrors validateEmailFromQuery(java.lang.String email)
```

POST method for validating a user's email address.

This method should be used to check only a specific field; other fields (if the request contains an XML- or JSON-encoded user object) are ignored in validation.

To create a user and validate all fields, use the [/validation/user](#) path ([validateUser\(String, String\)](#)).

This method is most commonly used with a form-encoded parameter, like
`emailaddress=[emailaddress_to_validate]`.

Examples:

Form-encoded user validation request

```
curl -d 'emailaddress=[emailaddress_to_validate]' \  
--header "authorization:Basic EncodedExampleAuthorization" \  
--header "content-type:x-www-form-urlencoded" \  
https://<authentication-host>:<port>/validation/user/emailaddress
```

`emailaddress_to_validate`

- The potential email address as a string

`<EncodedExampleAuthorization>`

- The basic HTTP authentication header parameter

`<authentication-host>`

- The host that URS API is running on

`<port>`

- The port number of the URS API

`Response`

Errors that occurred in validation (via XML representation of [ValidationErrors](#)) or an empty response on success.

| Status Code | Code Description | Possible Error Conditions |
|-------------|------------------|--|
| 200 | OK | Validation was performed, but errors were found and are listed in body of the response |
| 204 | No Content | Validation was performed and no errors were found |

Parameters:

`email` - the String to validate as an email address

Returns:

Overview (User Registration System API 2.55 API)

null if there are no validation errors; otherwise, a `ValidationErrors` object containing a list of `ValidationErrors` representing problems that occurred during validation of the supplied data.

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api Class ValidResource

```
java.lang.Object
└ gov.nasa.eosdis.urs.api.ValidResource
```

```
public class ValidResource
extends java.lang.Object
```

Server side REST interface for requesting lists of valid values for a given property. These properties are pre-defined and known on the URS system as well as the client system.

Note that this resource does *not* perform validation. Rather, it returns a list of valid values and allows individual applications to handle validation in their own way.

Endpoints:

```
<root-url>/valids/[valid-property-identifier]
```

- ◊ All the requests will be directed to this URL.
- ◊ This end point does not require or accept any input data.
- ◊ Additional HTML header parameters are authentication
 - (required) the requesting authority ([basic HTTP authentication](#))

Constructor Summary

| | |
|--|--|
| <u>ValidResource()</u> | |
|--|--|

Method Summary

| | |
|---|--|
| gov.nasa.eosdis.urs.core.objects.Valids | <u>getAffiliations()</u> GET method for requesting a list of acceptable (valid) affiliation values. |
| gov.nasa.eosdis.urs.core.objects.Valids | <u>getCountries()</u> GET method for requesting a list of acceptable (valid) countries. |
| gov.nasa.eosdis.urs.core.objects.Valids | <u>getStudyAreas()</u> GET method for requesting a list of acceptable (valid) primary study areas. |
| gov.nasa.eosdis.urs.core.objects.Valids | <u>getUserTypes()</u> GET method for requesting a list of acceptable (valid) user types. |

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

ValidsResource

```
public ValidsResource()
```

Method Detail

getCountries

```
public gov.nasa.eosdis.urs.core.objects.Valids getCountries()
```

GET method for requesting a list of acceptable (valid) countries. No input data is required or accepted as part of this request.

Example:

Request to get all acceptable affiliation values (most likely from a client-side Ajax request).

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
      https://<authentication-host>:<port>/valids/countries
```

<EncodedExampleAuthorization>
 - The basic HTTP authentication header parameter
<authentication-host>
 - The host that URS API is running on
<port>
 - The port number of the URS API

Response

an XML representation of a valids list containing the permitted country values.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Valids retrieval request was accepted and processed. The client must check the response contents to confirm whether or not the authentication was permitted. If not, the response will contain an error message giving the reason for the denial. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |

Returns:

an XML representation of a valids list containing the permitted country values.

getAffiliations

```
public gov.nasa.eosdis.urs.core.objects.Valids getAffiliations()
```

GET method for requesting a list of acceptable (valid) affiliation values. No input data is required or accepted as part of this request.

Example:

Request to get all acceptable affiliation values (most likely from a client-side Ajax request).

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
      https://<authentication-host>:<port>/valids/affiliations
```

<EncodedExampleAuthorization>

- The basic HTTP authentication header parameter

<authentication-host>

- The host that URS API is running on

<port>

- The port number of the URS API

Response

an XML representation of a valids list containing the permitted affiliation values.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Valids retrieval request was accepted and processed. The client must check the response contents to confirm whether or not the authentication was permitted. If not, the response will contain an error message giving the reason for the denial. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |

Returns:

an XML representation of a valids list containing the permitted affiliation values.

getUserTypes

```
public gov.nasa.eosdis.urs.core.objects.Valids getUserTypes()
```

GET method for requesting a list of acceptable (valid) user types. No input data is required or accepted as part of this request.

Example:

Request to get all acceptable user types (most likely from a client-side Ajax request).

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
      https://<authentication-host>:<port>/valids/usertypes
```

<EncodedExampleAuthorization>

- The basic HTTP authentication header parameter

<authentication-host>

- The host that URS API is running on

<port>

- The port number of the URS API

Overview (User Registration System API 2.55 API)

Response

an XML representation of a valids list containing the permitted user type values.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Valids retrieval request was accepted and processed. The client must check the response contents to confirm whether or not the authentication was permitted. If not, the response will contain an error message giving the reason for the denial. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |

Returns:

an XML representation of a valids list containing the permitted user type values.

getStudyAreas

```
public gov.nasa.eosdis.urs.core.objects.Valids getStudyAreas()
```

GET method for requesting a list of acceptable (valid) primary study areas. No input data is required or accepted as part of this request.

Example:

Request to get all acceptable primary study areas (most likely from a client-side Ajax request).

```
curl --header "authorization:Basic <EncodedExampleAuthorization>" \
      https://<authentication-host>:<port>/valids/studyareas
```

```
<EncodedExampleAuthorization>
  - The basic HTTP authentication header parameter
<authentication-host>
  - The host that URS API is running on
<port>
  - The port number of the URS API
```

Response

an XML representation of a valids list containing the permitted primary study area values.

| Status Code | Code Description | Possible Error Conditions |
|-------------|-----------------------|---|
| 200 | OK | Valids retrieval request was accepted and processed. The client must check the response contents to confirm whether or not the authentication was permitted. If not, the response will contain an error message giving the reason for the denial. |
| 500 | Internal Server Error | This should only occur as a result of system level errors, such as a system configuration problem. |

Returns:

an XML representation of a valids list containing the permitted primary study area values.

gov.nasa.eosdis.urs.api.exceptions

Class BadRequestException

```

java.lang.Object
  └─ java.lang.Throwable
    └─ java.lang.Exception
      └─ java.lang.RuntimeException
        └─ javax.ws.rs.WebApplicationException
          └─ gov.nasa.eosdis.urs.api.exceptions.BadRequestException

```

All Implemented Interfaces:

java.io.Serializable

```

public class BadRequestException
extends javax.ws.rs.WebApplicationException

```

See Also:

[Serialized Form](#)

Constructor Summary

[BadRequestException](#)(java.lang.String message)

Method Summary

| | |
|------------------|--------------------------------------|
| java.lang.String | <u>getMessage</u> () |
|------------------|--------------------------------------|

| | |
|---------------------------|---------------------------------------|
| javax.ws.rs.core.Response | <u>getResponse</u> () |
|---------------------------|---------------------------------------|

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`,
`printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

BadRequestException

```
public BadRequestException(java.lang.String message)
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

getResponse in class javax.ws.rs.WebApplicationException

getMessage

```
public java.lang.String getMessage()
```

Overrides:

getMessage in class java.lang.Throwable

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.exceptions

Class ConflictErrorException

```

java.lang.Object
  ↘ java.lang.Throwable
    ↘ java.lang.Exception
      ↘ java.lang.RuntimeException
        ↘ javax.ws.rs.WebApplicationException
          ↘ gov.nasa.eosdis.urs.api.exceptions.ConflictErrorException

```

All Implemented Interfaces:

java.io.Serializable

```

public class ConflictErrorException
extends javax.ws.rs.WebApplicationException

```

See Also:

[Serialized Form](#)

Constructor Summary

ConflictErrorException(java.lang.String message)

Method Summary

| | |
|--|--|
| | java.lang.String <u>getMessage</u> () |
|--|--|

| | |
|--|--|
| | javax.ws.rs.core.Response <u>getResponse</u> () |
|--|--|

Methods inherited from class java.lang.Throwable

| |
|---|
| fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString |
|---|

Methods inherited from class java.lang.Object

| |
|---|
| equals, getClass, hashCode, notify, notifyAll, wait, wait, wait |
|---|

Constructor Detail

ConflictErrorException

```
public ConflictErrorException(java.lang.String message)
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

getResponse in class javax.ws.rs.WebApplicationException

getMessage

```
public java.lang.String getMessage()
```

Overrides:

getMessage in class java.lang.Throwable

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.exceptions Class ForbiddenException

```
java.lang.Object
└ java.lang.Throwable
  └ java.lang.Exception
    └ java.lang.RuntimeException
      └ javax.ws.rs.WebApplicationException
        └ gov.nasa.eosdis.urs.api.exceptions.ForbiddenException
```

All Implemented Interfaces:

java.io.Serializable

```
public class ForbiddenException
extends javax.ws.rs.WebApplicationException
```

See Also:

[Serialized Form](#)

Constructor Summary

[**ForbiddenException\(\)**](#)

[**ForbiddenException\(java.lang.String message\)**](#)

Method Summary

| | |
|---------------------------|--------------------------------------|
| javax.ws.rs.core.Response | getResponse() |
|---------------------------|--------------------------------------|

Methods inherited from class `java.lang.Throwable`

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ForbiddenException

```
public ForbiddenException(java.lang.String message)
```

ForbiddenException

```
public ForbiddenException()
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

getResponse in class javax.ws.rs.WebApplicationException

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.exceptions Class InternalServerErrorException

```
java.lang.Object
  └─ java.lang.Throwable
    └─ java.lang.Exception
      └─ java.lang.RuntimeException
        └─ javax.ws.rs.WebApplicationException
          └─ gov.nasa.eosdis.urs.api.exceptions.InternalServerErrorException
```

All Implemented Interfaces:

java.io.Serializable

```
public class InternalServerErrorException
extends javax.ws.rs.WebApplicationException
```

See Also:

[Serialized Form](#)

Constructor Summary

[InternalServerErrorException\(\)](#)

[InternalServerErrorException\(java.lang.Exception e\)](#)

[InternalServerErrorException\(java.lang.String message\)](#)

[InternalServerErrorException\(java.lang.String message, java.lang.Exception e\)](#)

Method Summary

| | |
|---------------------------|--------------------------------------|
| javax.ws.rs.core.Response | <u>getResponse()</u> |
|---------------------------|--------------------------------------|

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace,
initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace,
toString`

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

InternalServerException

```
public InternalServerException()
```

InternalServerException

```
public InternalServerException(java.lang.String message)
```

InternalServerException

```
public InternalServerException(java.lang.Exception e)
```

InternalServerException

```
public InternalServerException(java.lang.String message,  
                               java.lang.Exception e)
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

getResponse in class javax.ws.rs.WebApplicationException

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.exceptions

Class ServiceUnavailableException

```

java.lang.Object
  ↘ java.lang.Throwable
    ↘ java.lang.Exception
      ↘ java.lang.RuntimeException
        ↘ javax.ws.rs.WebApplicationException
          ↘ gov.nasa.eosdis.urs.api.exceptions.ServiceUnavailableException

```

All Implemented Interfaces:

java.io.Serializable

```

public class ServiceUnavailableException
extends javax.ws.rs.WebApplicationException

```

See Also:

[Serialized Form](#)

Constructor Summary

[ServiceUnavailableException\(\)](#)

[ServiceUnavailableException\(java.lang.String message\)](#)

Method Summary

| | |
|---------------------------|--------------------------------------|
| javax.ws.rs.core.Response | <u>getResponse()</u> |
|---------------------------|--------------------------------------|

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ServiceUnavailableException

```
public ServiceUnavailableException()
```

ServiceUnavailableException

```
public ServiceUnavailableException(java.lang.String message)
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

getResponse in class javax.ws.rs.WebApplicationException

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.exceptions Class UnauthorizedException

```
java.lang.Object
└ java.lang.Throwable
  └ java.lang.Exception
    └ java.lang.RuntimeException
      └ javax.ws.rs.WebApplicationException
        └ gov.nasa.eosdis.urs.api.exceptions.UnauthorizedException
```

All Implemented Interfaces:

java.io.Serializable

```
public class UnauthorizedException
extends javax.ws.rs.WebApplicationException
```

Exception to support basic http authentication.

See Also:

[Serialized Form](#)

Constructor Summary

[UnauthorizedException\(\)](#)

[UnauthorizedException\(java.lang.String message\)](#)

Method Summary

| | |
|---------------------------|--------------------------------------|
| javax.ws.rs.core.Response | <u>getResponse()</u> |
|---------------------------|--------------------------------------|

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

UnauthorizedException

```
public UnauthorizedException()
```

UnauthorizedException

```
public UnauthorizedException(java.lang.String message)
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

```
getResponse in class javax.ws.rs.WebApplicationException
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.exceptions Class ValidationErrorsException

```
java.lang.Object
  └─ java.lang.Throwable
    └─ java.lang.Exception
      └─ java.lang.RuntimeException
        └─ javax.ws.rs.WebApplicationException
          └─ gov.nasa.eosdis.urs.api.exceptions.ValidationErrorsException
```

All Implemented Interfaces:

java.io.Serializable

```
public class ValidationErrorsException
extends javax.ws.rs.WebApplicationException
```

See Also:

[Serialized Form](#)

Constructor Summary

| |
|---|
| <u>ValidationErrorsException</u> (java.util.Collection<ValidationErrors> errors) |
| <u>ValidationErrorsException</u> (java.lang.String field, java.lang.String message) |
| <u>ValidationErrorsException</u> (ValidationErrors errors) |

Method Summary

| | |
|---------------------------|---------------------------------------|
| java.lang.String | <u>getMessage</u> () |
| javax.ws.rs.core.Response | <u>getResponse</u> () |

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`,
`printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

ValidationErrorsException

```
public ValidationErrorsException(ValidationErrors errors)
```

ValidationErrorsException

```
public ValidationErrorsException(java.util.Collection<ValidationError> errors)
```

ValidationErrorsException

```
public ValidationErrorsException(java.lang.String field,  
                                java.lang.String message)
```

Method Detail

getResponse

```
public javax.ws.rs.core.Response getResponse()
```

Overrides:

getResponse in class javax.ws.rs.WebApplicationException

getMessage

```
public java.lang.String getMessage()
```

Overrides:

getMessage in class java.lang.Throwable

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.objects

Class GroupList

```
java.lang.Object
└ gov.nasa.eosdis.urs.api.objects.GroupList
```

```
public class GroupList
extends java.lang.Object
```

Field Summary

| | |
|--|---------------|
| java.util.List<gov.nasa.eosdis.urs.core.objects.GroupInfo> | groups |
|--|---------------|

Constructor Summary

GroupList()

Class constructor.

GroupList(java.util.List<gov.nasa.eosdis.urs.core.objects.GroupInfo> groups)

Class constructor.

Method Summary

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

groups

```
public java.util.List<gov.nasa.eosdis.urs.core.objects.GroupInfo> groups
```

Constructor Detail

GroupList

```
public GroupList()
```

Class constructor.

GroupList

```
public GroupList(java.util.List<gov.nasa.eosdis.urs.core.objects.GroupInfo> groups)
```

Class constructor.

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.objects

Class LoginRequest

```
java.lang.Object
└ gov.nasa.eosdis.urs.api.objects.LoginRequest
```

```
public class LoginRequest
extends java.lang.Object
```

Constructor Summary

[LoginRequest \(\)](#)

[LoginRequest \(java.lang.String userName, java.lang.String password\)](#)

[LoginRequest \(java.lang.String userName, java.lang.String password,](#)
[gov.nasa.eosdis.urs.core.objects.UserFields userFields\)](#)

Method Summary

| | |
|---|---|
| java.lang.String | <u>getPassword ()</u> |
| gov.nasa.eosdis.urs.core.objects.UserFields | <u>getUserFields ()</u> |
| java.lang.String | <u>getUsername ()</u> |
| void | <u>setPassword (java.lang.String password)</u> |
| void | <u>setUserFields (gov.nasa.eosdis.urs.core.objects.UserFields userFields)</u> |
| void | <u> setUsername (java.lang.String username)</u> |
| java.lang.String | <u>toString ()</u> |

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

LoginRequest

```
public LoginRequest()
```

LoginRequest

```
public LoginRequest(java.lang.String userName,  
                   java.lang.String password)
```

LoginRequest

```
public LoginRequest(java.lang.String userName,  
                   java.lang.String password,  
                   gov.nasa.eosdis.urs.core.objects.UserFields userFields)
```

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:

toString in class java.lang.Object

setUsername

```
public void setUsername(java.lang.String username)
```

Parameters:

username - the username to set

getUsername

```
public java.lang.String getUsername()
```

Returns:

the username

setPassword

```
public void setPassword(java.lang.String password)
```

Parameters:

password - the password to set

getPassword

```
public java.lang.String getPassword()
```

Returns:

the password

setUserFields

```
public void setUserFields(gov.nasa.eosdis.urs.core.objects.UserFields userFields)
```

getUserFields

```
public gov.nasa.eosdis.urs.core.objects.UserFields getUserFields()
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.objects

Class ValidationError

```
java.lang.Object
└ gov.nasa.eosdis.urs.api.objects.ValidationError
```

```
public class ValidationError
extends java.lang.Object
```

Field Summary

| | |
|------------------|--------------------------------|
| java.lang.String | <u>field</u> |
| java.lang.String | <u>message</u> |

Constructor Summary

[ValidationError\(\)](#)

[ValidationError\(java.lang.String field, java.lang.String error\)](#)

Method Summary

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

field

```
public java.lang.String field
```

message

```
public java.lang.String message
```

Constructor Detail

ValidationError

```
public ValidationError()
```

ValidationError

```
public ValidationError(java.lang.String field,  
                      java.lang.String error)
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.api.objects

Class ValidationErrors

```
java.lang.Object
└ gov.nasa.eosdis.urs.api.objects.ValidationErrors
```

```
public class ValidationErrors
extends java.lang.Object
```

Field Summary

| | |
|--|-------------------------|
| java.util.Collection<ValidationErrors> | validationErrors |
|--|-------------------------|

Constructor Summary

ValidationErrors()

ValidationErrors(java.util.Collection<ValidationErrors> validationErrors)

Method Summary

Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

validationErrors

```
public java.util.Collection<ValidationErrors> validationErrors
```

Constructor Detail

ValidationErrors

```
public ValidationErrors()
```

ValidationErrors

```
public ValidationErrors(java.util.Collection<ValidationError> validationErrors)
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.extendedwadl

Class URSMODELEXAMPLES

```
java.lang.Object
└ gov.nasa.eosdis.urs.extendedwadl.URSMODELEXAMPLES
```

```
public class URSMODELEXAMPLES
extends java.lang.Object
```

This class provides example representations that can be used for wadl docs.

Author:

kbaynes

Field Summary

| | |
|--|---|
| static gov.nasa.eosdis.urs.core.objects.Group | <u>SAMPLE GROUP</u> |
| static <u>LoginRequest</u> | <u>SAMPLE LOGIN REQUEST</u> |
| static gov.nasa.eosdis.urs.core.objects.SearchResponse | <u>SAMPLE SEARCH RESPONSE</u> |
| static gov.nasa.eosdis.urs.core.objects.User | <u>SAMPLE USER</u> |
| static <u>ValidationErrors</u> | <u>SAMPLE VALIDATION ERROR</u> |
| static <u>ValidationErrors</u> | <u>SAMPLE VALIDATION ERRORS</u> |

Constructor Summary

[URSMODELEXAMPLES\(\)](#)

Method Summary

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

SAMPLE_USER

```
public static final gov.nasa.eosdis.urs.core.objects.User SAMPLE_USER
```

SAMPLE_SEARCH_RESPONSE

```
public static final gov.nasa.eosdis.urs.core.objects.SearchResponse SAMPLE_SEARCH_RESPONSE
```

SAMPLE_LOGIN_REQUEST

```
public static final LoginRequest SAMPLE_LOGIN_REQUEST
```

SAMPLE_VALIDATION_ERROR

```
public static final ValidationErrors SAMPLE_VALIDATION_ERROR
```

SAMPLE_VALIDATION_ERRORS

```
public static final ValidationErrors SAMPLE_VALIDATION_ERRORS
```

SAMPLE_GROUP

```
public static final gov.nasa.eosdis.urs.core.objects.Group SAMPLE_GROUP
```

Constructor Detail

URSModelExamples

```
public URSModelExamples()
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util Class AccountVerifier

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.AccountVerifier
```

```
public class AccountVerifier
extends java.lang.Object
```

Field Summary

| | |
|-------------------------|--------------------------|
| static java.lang.String | <u>ACTIVATION</u> |
|-------------------------|--------------------------|

| | |
|-------------------------|----------------------------|
| static java.lang.String | <u>VERIFICATION</u> |
|-------------------------|----------------------------|

Constructor Summary

AccountVerifier(java.lang.String key, java.lang.String type)
Class constructor.

AccountVerifier(gov.nasa.eosdis.urs.core.objects.User user, java.lang.String app)
Class constructor.

AccountVerifier(gov.nasa.eosdis.urs.core.objects.User user, java.lang.String app,
java.lang.String type)
Class constructor.

Method Summary

| | |
|------------------|--|
| java.lang.String | <u>getApp()</u> Returns the application that requested creation of this account. |
|------------------|--|

| | |
|------------------|--|
| java.lang.String | <u>getKey()</u> Returns the activation key for this account. |
|------------------|--|

| | |
|---------------------------------------|--|
| gov.nasa.eosdis.urs.core.objects.User | <u>getUser()</u> Returns the user account details. |
|---------------------------------------|--|

| | |
|------|--|
| void | <u>sendMail</u> (java.lang.String baseuri, java.lang.String guiuri, java.lang.String type) Send mail. |
|------|--|

| | |
|------|---|
| void | <u>sendMail</u> (java.lang.String baseuri, java.lang.String guiuri, java.lang.String address, java.lang.String type) Sends an email with an account activation link embedded into it. |
|------|---|

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail**ACTIVATION**

```
public static final java.lang.String ACTIVATION
```

See Also:

[Constant Field Values](#)

VERIFICATION

```
public static final java.lang.String VERIFICATION
```

See Also:

[Constant Field Values](#)

Constructor Detail**AccountVerifier**

```
public AccountVerifier(gov.nasa.eosdis.urs.core.objects.User user,
                      java.lang.String app)
```

Class constructor. This constructor will generate an application key for the given user and application.

Parameters:

user - the user account to create an verification key for
app - the application that requested the account creation

AccountVerifier

```
public AccountVerifier(gov.nasa.eosdis.urs.core.objects.User user,
                      java.lang.String app,
                      java.lang.String type)
```

Class constructor. This constructor will generate an application key for the given user and application.

Parameters:

user - the user account to create an verification key for
app - the application that requested the account creation

AccountVerifier

```
public AccountVerifier(java.lang.String key,
                      java.lang.String type)
```

Class constructor. This constructor will obtain the user information for a given account activation request.

Parameters:

key - the activation key

Method Detail

sendMail

```
public void sendMail(java.lang.String baseuri,
                     java.lang.String guiuri,
                     java.lang.String address,
                     java.lang.String type)
```

Sends an email with an account activation link embedded into it.

Parameters:

baseuri - the baseuri of the server generating this request.
 guiuri - the guiuri
 address - the email address to be sent to
 type - the type

sendMail

```
public void sendMail(java.lang.String baseuri,
                     java.lang.String guiuri,
                     java.lang.String type)
```

Send mail. This sends the email to the user's email address

Parameters:

baseuri - the baseuri
 guiuri - the guiuri
 type - the type

getUser

```
public gov.nasa.eosdis.urs.core.objects.User getUser()
```

Returns the user account details.

Returns:

the user object for the account

getApp

```
public java.lang.String getApp()
```

Returns the application that requested creation of this account. This may be null. If not null, the API should redirect an activation caller to the application redirection page.

Returns:

the application ID, or null if none.

getKey

```
public java.lang.String getKey()
```

Returns the activation key for this account. This is in the form of a hexadecimal string

Returns:

the verification key.

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util Class AuthorityValidation

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.AuthorityValidation
```

```
public class AuthorityValidation
extends java.lang.Object
```

Constructor Summary

[AuthorityValidation](#)(gov.nasa.eosdis.urs.core.common.ChangeAuthority authority)

Method Summary

| | |
|---------|---|
| boolean | <u>isAuthorityValid()</u> |
|---------|---|

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

AuthorityValidation

```
public AuthorityValidation(gov.nasa.eosdis.urs.core.common.ChangeAuthority authority)
```

Method Detail

isAuthorityValid

```
public boolean isAuthorityValid()
    throws gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException,
           gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException,
           gov.nasa.eosdis.urs.core.exception.InvalidValueException,
           gov.nasa.eosdis.urs.core.dao.exception.SearchException,
           gov.nasa.eosdis.urs.core.dao.exception.DaoActionException,
           com.unboundid.ldap.sdk.LDAPSearchException
```

Overview (User Registration System API 2.55 API)

Throws:

```
gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException  
gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException  
gov.nasa.eosdis.urs.core.exception.InvalidValueException  
gov.nasa.eosdis.urs.core.dao.exception.SearchException  
gov.nasa.eosdis.urs.core.dao.exception.DaoActionException  
com.unboundid.ldap.sdk.LDAPSearchException
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util Class GroupUtils

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.GroupUtils
```

```
public class GroupUtils
extends java.lang.Object
```

Author:

mhuffnagle

Constructor Summary

[GroupUtils\(\)](#)

Method Summary

| | |
|---|---|
| static boolean | <u>isValidGroupName</u> (java.lang.String name) |
| Determines if a given group name is valid | |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

GroupUtils

```
public GroupUtils()
```

Method Detail

isValidGroupName

```
public static boolean isValidGroupName(java.lang.String name)
```

Determines if a given group name is valid

Parameters:

Overview (User Registration System API 2.55 API)

name - the name to test

Returns:

true if the name is valid, false otherwise

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util

Class Mailer

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.Mailer
```

```
public class Mailer
extends java.lang.Object
```

Utility class used to compose emails for sending.

Constructor Summary

[Mailer\(\)](#)

Method Summary

| | |
|-------------------------|--|
| static java.lang.String | <u>generateMail</u> (java.lang.String mailname, java.util.HashMap<java.lang.String, java.lang.String> parameters) Generates an email body from a named resource file and a series of parameters. |
| static void | <u>send</u> (java.lang.String to, java.lang.String from, java.lang.String subject, java.lang.String body) Writes an email into a file for processing by the mailer script. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Mailer

```
public Mailer()
```

Method Detail

send

```
public static void send(java.lang.String to,  
                      java.lang.String from,  
                      java.lang.String subject,  
                      java.lang.String body)
```

Writes an email into a file for processing by the mailer script.

Parameters:

- to -
- from -
- subject -
- body -

generateMail

```
public static java.lang.String generateMail(java.lang.String mailname,  
                                         java.util.HashMap<java.lang.String, java.lang.String> parameters)  
throws gov.nasa.eosdis.urs.core.common.UrsException
```

Generates an email body from a named resource file and a series of parameters.

Parameters:

- mailname - the name of the email file. This does not include the ".txt" suffix, or the "mail/" prefix.
- parameters - name/value pairs to be inserted into the email body text.

Returns:

- the string representing the body of the email.

Throws:

- gov.nasa.eosdis.urs.core.common.UrsException - if the named resource (email file) could not be found or read.

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util Class PasswordUtils

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.PasswordUtils
```

```
public class PasswordUtils
extends java.lang.Object
```

Constructor Summary

[PasswordUtils\(\)](#)

Method Summary

| | |
|-------------------------|--|
| static boolean | <u>isPasswordEncrypted</u> (java.lang.String password) Checks to see whether a user password is encrypted |
| static boolean | <u>isPasswordPlaintext</u> (java.lang.String password) Verifies that the password does not have an encryption prefix. |
| static java.lang.String | <u>randomPassword</u> (int min, int max) Generates a random password with upper and lower constraints on its length. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

PasswordUtils

```
public PasswordUtils()
```

Method Detail

isPasswordEncrypted

```
public static boolean isPasswordEncrypted(java.lang.String password)
```

Overview (User Registration System API 2.55 API)

Checks to see whether a user password is encrypted

Parameters:

password - the password to test for encryption

Returns:

true if the password is encrypted with a recognizable algorithm, false otherwise.

isPasswordPlaintext

```
public static boolean isPasswordPlaintext(java.lang.String password)
```

Verifies that the password does not have an encryption prefix.

Parameters:

password - the password to verify.

randomPassword

```
public static java.lang.String randomPassword(int min,  
                                             int max)
```

Generates a random password with upper and lower constraints on its length.

Parameters:

min - the minimum length of the password

max - the maximum length of the password

Returns:

a newly created password

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util**Class Randomizer**

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.Randomizer
```

```
public class Randomizer
extends java.lang.Object
```

Randomizer is a pseudo random number generator that builds on java.util.Random and allows multiple instances to share a single instance of that class, which should be more random than creating a new Random instance every time one is needed. All tests and data generators should use this class unless there is a good reason not to. If a repeatable string of randomness is needed, use the constructor that takes a seed. Each instance created with that constructor will have its own Random instance, initialized with the given seed.

Constructor Summary**Randomizer()**

Creates a new Randomizer instance that uses the singleton random number generator shared by all other Randomizer instances created with this constructor.

Randomizer(long seed)

Creates a new Randomizer with the given seed for repeatable pseudorandomness.

Method Summary

| | |
|-------|---|
| <T> T | <u>choose</u> (java.util.List<T> items) Randomly selects an item from a list. |
|-------|---|

| | |
|-------|--|
| <T> T | <u>choose</u> (T... items) Randomly selects an item from an array. |
|-------|--|

| | |
|-------------|--|
| static void | <u>main</u> (java.lang.String[] args) |
|-------------|--|

| | |
|---------|------------------------------|
| boolean | <u>nextBoolean</u> () |
|---------|------------------------------|

| | |
|------|---|
| void | <u>nextBytes</u> (byte[] bytes) Generates random bytes and places them into a user-supplied byte array. |
|------|---|

| | |
|--------|-----------------------------|
| double | <u>nextDouble</u> () |
|--------|-----------------------------|

| | |
|-------|----------------------------|
| float | <u>nextFloat</u> () |
|-------|----------------------------|

| | |
|--------|-------------------------------|
| double | <u>nextGaussian</u> () |
|--------|-------------------------------|

Overview (User Registration System API 2.55 API)

| | |
|-----------------------|--|
| | int <u>nextInt</u> () |
| | int <u>nextInt</u> (int upperBound) |
| | int <u>nextInt</u> (int min, int max) Returns a randomly chosen int between 'min' inclusive and 'max' exclusive. |
| | long <u>nextLong</u> () |
| | long <u>nextLong</u> (long upperBound) |
| java.lang.String | <u>random</u> (java.lang.String source, int length) Creates a String of 'length' characters randomly chosen from 'source'. |
| java.lang.String | <u>random</u> (java.lang.String source, int min, int max) Randomly selects characters from 'source' to build a new String whose length is between 'min' and 'max', inclusive. |
| java.lang.String | <u>randomPassword</u> (int length) Generates a password of upper and lowercase letters and digits, where each character is equally likely to be from each of the 3 character sets. |
| java.lang.String | <u>randomPassword</u> (int length, java.lang.String... charSets) Generates a random string from the given character sets, so that each character is equally likely to come from any character set, resulting in a different distribution than randomly choosing characters from one large character set. |
| void | <u>shuffle</u> (java.util.List<?> list) Randomly permutes a list destructively. |
| <T> void | <u>shuffle</u> (T[] array) Randomly permutes an array destructively. |
| <T> java.util.List<T> | <u>shuffled</u> (java.util.List<T> list) Randomly permutes a copy of list. |
| java.lang.String | <u>shuffled</u> (java.lang.String str) Returns a randomly permuted copy of str. |
| <T> T[] | <u>shuffled</u> (T[] array) Randomly permutes a copy of array. |

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Randomizer

```
public Randomizer()
```

Creates a new Randomizer instance that uses the singleton random number generator shared by all other Randomizer instances created with this constructor.

Randomizer

```
public Randomizer(long seed)
```

Creates a new Randomizer with the given seed for repeatable pseudorandomness.

Parameters:

seed - the initial seed.

Method Detail

shuffled

```
public <T> java.util.List<T> shuffled(java.util.List<T> list)
```

Randomly permutes a copy of list. Does not modify the original list. This should be preferred over the destructive version.

Type Parameters:

T - The type of elements in list.

Parameters:

list - A list of elements to shuffle.

Returns:

a randomly permuted copy of list.

shuffled

```
public <T> T[] shuffled(T[] array)
```

Randomly permutes a copy of array. Does not modify the original array. This should be preferred over the destructive version.

Type Parameters:

T - The type of elements in array.

Parameters:

array - An array of elements to shuffle.

Returns:

a randomly permuted copy of array.

shuffled

```
public java.lang.String shuffled(java.lang.String str)
```

Returns a randomly permuted copy of str.

Parameters:

str - A string to shuffle.

Returns:

A shuffled copy of str.

shuffle

```
public void shuffle(java.util.List<?> list)
```

Randomly permutes a list destructively.

Parameters:

list - The List to shuffle.

shuffle

```
public <T> void shuffle(T[] array)
```

Randomly permutes an array destructively.

Type Parameters:

T - The type of elements in the array

Parameters:

array - The array to shuffle.

nextBoolean

```
public boolean nextBoolean()
```

Returns:

the next pseudorandom, uniformly distributed boolean value from this random number generator's sequence.

nextDouble

```
public double nextDouble()
```

Returns:

the next pseudorandom, uniformly distributed double value between 0.0 and 1.0 from this random number generator's sequence.

nextFloat

```
public float nextFloat()
```

Returns:

the next pseudorandom, uniformly distributed float value between 0.0 and 1.0 from this random number generator's sequence.

nextGaussian

```
public double nextGaussian()
```

Returns:

the next pseudorandom, Gaussian ("normally") distributed double value with mean 0.0 and standard deviation 1.0 from this random number generator's sequence.

nextInt

```
public int nextInt()
```

Returns:

the next pseudorandom, uniformly distributed int value from this random number generator's sequence.

nextInt

```
public int nextInt(int upperBound)
```

Parameters:

upperBound - The bound on the random number to be returned. Must be positive.

Returns:

the next pseudorandom, uniformly distributed int value between 0 (inclusive) and upperBound (exclusive) from this random number generator's sequence.

nextLong

```
public long nextLong()
```

Returns:

the next pseudorandom, uniformly distributed long value from this random number generator's sequence.

nextLong

```
public long nextLong(long upperBound)
```

Parameters:

upperBound - The bound on the random number to be returned. Must be positive.

Returns:

the next pseudorandom, uniformly distributed long value between 0 (inclusive) and upperBound (exclusive) from this random number generator's sequence

nextBytes

```
public void nextBytes(byte[] bytes)
```

Generates random bytes and places them into a user-supplied byte array. The number of random bytes produced is equal to the length of the byte array.

Parameters:

bytes - the byte array to fill with random bytes

Throws:

java.lang.NullPointerException - if the byte array is null

See Also:

Random

choose

```
public <T> T choose(T... items)
```

Randomly selects an item from an array.

Type Parameters:

T - The type of items in the array.

Parameters:

items - An array of things.

Returns:

a randomly selected item from the array.

choose

```
public <T> T choose(java.util.List<T> items)
```

Randomly selects an item from a list.

Type Parameters:

T - The type of items in the list.

Parameters:

items - A list of things.

Returns:

a randomly selected item from the list.

random

```
public java.lang.String random(java.lang.String source,  
                           int length)
```

Creates a String of 'length' characters randomly chosen from 'source'.

Parameters:

source - A String of characters to use in building the random String.
length - The length of the new String.

Returns:

A String of 'length' characters randomly chosen from 'source'.

randomPassword

```
public java.lang.String randomPassword(int length)
```

Generates a password of upper and lowercase letters and digits, where each character is equally likely to be from each of the 3 character sets. This is a different distribution than choosing randomly from the ALNUM character set, in which there are 26 chances to get an uppercase letter, 26 chances to get a lowercase letter, and 10 chances to get a digit.

Parameters:

length - the password length.

Returns:

a randomly generated password.

randomPassword

```
public java.lang.String randomPassword(int length,  
                                      java.lang.String... charSets)
```

Generates a random string from the given character sets, so that each character is equally likely to come from any character set, resulting in a different distribution than randomly choosing characters from one large character set. The distribution will be skewed if the character sets are different sizes.

Parameters:

length - the password length.
charSets - the sets of characters from which to build the password.

Returns:

a randomly generated password.

random

```
public java.lang.String random(java.lang.String source,  
                               int min,  
                               int max)
```

Randomly selects characters from 'source' to build a new String whose length is between 'min' and 'max', inclusive.

Parameters:

source - A String of characters to use in building the random String.
min - The minimum length of the new String.
max - The maximum length of the new String.

Returns:

A String of from 'min' to 'max' characters randomly chosen from 'source'.

Throws:

java.lang.IllegalArgumentException - if min is not less than max.

nextInt

```
public int nextInt(int min,  
                   int max)
```

Returns a randomly chosen int between 'min' inclusive and 'max' exclusive.

Parameters:

- min - The inclusive lower bound.
- max - The exclusive upper bound.

Returns:

- an int 'n', where (min <= n < max).

Throws:

- java.lang.IllegalArgumentException - if min is not less than max.

main

```
public static void main(java.lang.String[] args)
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util Class ResourceLoader

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.ResourceLoader
```

```
public class ResourceLoader
extends java.lang.Object
```

Constructor Summary

[ResourceLoader\(\)](#)

Method Summary

| | |
|-------------------------|---|
| static java.lang.String | <u>loadResource</u> (java.lang.String filename) Loads a named resource file. |
|-------------------------|---|

| | |
|---|---|
| static java.util.List<java.lang.String> | <u>loadResourceList</u> (java.lang.String filename) Loads a named resource file. |
|---|---|

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

ResourceLoader

```
public ResourceLoader()
```

Method Detail

loadResourceList

```
public static java.util.List<java.lang.String> loadResourceList(java.lang.String filename)
throws gov.nasa.eosdis.urs.core.common.UrsException
```

Loads a named resource file.

Parameters:

Overview (User Registration System API 2.55 API)

filename - the name of the resource file to load.

Returns:

A list of strings containing the resource files contents.

Throws:

gov.nasa.eosdis.urs.core.common.UrsException - on error

loadResource

```
public static java.lang.String loadResource(java.lang.String filename)
                                         throws gov.nasa.eosdis.urs.core.common.UrsException
```

Loads a named resource file. Note that this method will translate line feed/carriage return pairs into a single line feed.

Parameters:

filename - the name of the resource file to load.

Returns:

the contents of the file as a string

Throws:

gov.nasa.eosdis.urs.core.common.UrsException - if the named resource could not be found, or
read.

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util Class StringConstants

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.StringConstants
```

```
public class StringConstants
extends java.lang.Object
```

Field Summary

| | |
|-------------------------|---|
| static java.lang.String | <u>ALNUM</u> The set of letters and numbers. |
| static java.lang.String | <u>ALPHA</u> The set of upper- and lowercase letters. |
| static java.lang.String | <u>DIGITS</u> The set of numeric characters. |
| static java.lang.String | <u>HEX_LOWER</u> Lowercase hexadecimal digits. |
| static java.lang.String | <u>HEX_UPPER</u> Uppercase hexadecimal digits. |
| static java.lang.String | <u>LOWER</u> The set of lowercase letters. |
| static java.lang.String | <u>PUNCT</u> Punctuation characters. |
| static java.lang.String | <u>UPPER</u> String classes for generating values composed of specific sets of characters (e.g., all letters, all numbers, hex digits). |

Method Summary

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

UPPER

```
public static final java.lang.String UPPER
```

The set of uppercase letters.

See Also:

[Constant Field Values](#)

LOWER

```
public static final java.lang.String LOWER
```

The set of lowercase letters.

ALPHA

```
public static final java.lang.String ALPHA
```

The set of upper- and lowercase letters.

DIGITS

```
public static final java.lang.String DIGITS
```

The set of numeric characters.

See Also:

[Constant Field Values](#)

ALNUM

```
public static final java.lang.String ALNUM
```

The set of letters and numbers.

HEX_UPPER

```
public static final java.lang.String HEX_UPPER
```

Uppercase hexadecimal digits.

See Also:

[Constant Field Values](#)

HEX_LOWER

```
public static final java.lang.String HEX_LOWER
```

Lowercase hexadecimal digits.

PUNCT

```
public static final java.lang.String PUNCT
```

Punctuation characters.

See Also:

[Constant Field Values](#)

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util**Class UserUtils**

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.UserUtils
```

```
public class UserUtils
extends java.lang.Object
```

Constructor Summary[UserUtils\(\)](#)**Method Summary**

| | |
|--|---|
| static gov.nasa.eosdis.urs.core.objects.User | <u>findUser</u> (gov.nasa.eosdis.urs.core.dao.UserDAO userDao, gov.nasa.eosdis.urs.core.common.ChangeAuthority authority, java.lang.String uid) Finds the user in LDAP by user name (uid). |
| static gov.nasa.eosdis.urs.core.objects.User | <u>findUserByEmail</u> (gov.nasa.eosdis.urs.core.dao.UserDAO userDao, gov.nasa.eosdis.urs.core.common.ChangeAuthority authority, java.lang.String email) Finds a user in LDAP by email address. |
| static gov.nasa.eosdis.urs.core.objects.User | <u>getPartialUser</u> (gov.nasa.eosdis.urs.core.objects.User user, java.util.List<java.lang.String> fields) Creates a partial user object from a full user object, using a list of desired attributes. |
| static java.lang.String | <u>jaxbObjectToString</u> (java.lang.Object obj) |
| static gov.nasa.eosdis.urs.core.objects.User | <u>xmlStringToUser</u> (java.lang.String xml) |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

UserUtils

```
public UserUtils()
```

Method Detail

findUser

```
public static gov.nasa.eosdis.urs.core.objects.User findUser(gov.nasa.eosdis.urs.core.dao.UserDAO userDao,
    gov.nasa.eosdis.urs.core.common.ChangeAuthority
    java.lang.String uid)
throws gov.nasa.eosdis.urs.core.exception.InvalidValueException,
gov.nasa.eosdis.urs.core.dao.exception.SearchException,
gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException,
gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException,
gov.nasa.eosdis.urs.core.dao.exception.DaoActionException
```

Finds the user in LDAP by user name (uid).

Parameters:

- userDao - the user DAO to use to search LDAP.
- authority - the credentials to use to search LDAP for the user.
- uid - the user's user name, or LDAP key.

Returns:

- the found user.

Throws:

- gov.nasa.eosdis.urs.core.dao.exception.DaoException - if there is an LDAP problem.
- gov.nasa.eosdis.urs.core.exception.InvalidValueException - the invalid value exception
- gov.nasa.eosdis.urs.core.dao.exception.DaoActionException
- gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException
- gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException
- gov.nasa.eosdis.urs.core.dao.exception.SearchException

findUserByEmail

```
public static gov.nasa.eosdis.urs.core.objects.User findUserByEmail(gov.nasa.eosdis.urs.core.dao.UserDAO user
    gov.nasa.eosdis.urs.core.common.ChangeAuthority
    java.lang.String email)
throws gov.nasa.eosdis.urs.core.dao.exception.SearchException,
gov.nasa.eosdis.urs.core.dao.exception.InvalidValueException,
gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException,
gov.nasa.eosdis.urs.core.dao.exception.DaoActionException,
gov.nasa.eosdis.urs.core.exception.InvalidValueException
```

Finds a user in LDAP by email address.

Parameters:

- userDao - the user DAO to use to search LDAP
- authority - the credentials to use for searching LDAP
- email - the email address to search for

Returns:

- the found user or null if no user is found

Overview (User Registration System API 2.55 API)

Throws:

```
gov.nasa.eosdis.urs.core.dao.exception.SearchException  
gov.nasa.eosdis.urs.core.dao.exception.ConnectivityException  
gov.nasa.eosdis.urs.core.dao.exception.InvalidAuthorityException  
gov.nasa.eosdis.urs.core.dao.exception.DaoActionException  
gov.nasa.eosdis.urs.core.exception.InvalidValueException
```

getPartialUser

```
public static gov.nasa.eosdis.urs.core.objects.User getPartialUser(gov.nasa.eosdis.urs.core.objects.User user,  
                      java.util.List<java.lang.String> fields)  
throws gov.nasa.eosdis.urs.core.dao.exception.Invali
```

Creates a partial user object from a full user object, using a list of desired user attributes.

Parameters:

user - the full user object
fields - a list of the desired fields

Returns:

a partial user object.

Throws:

```
gov.nasa.eosdis.urs.core.dao.exception.InvalidOperationException
```

jaxbObjectToString

```
public static java.lang.String jaxbObjectToString(java.lang.Object obj)  
throws javax.xml.bind.JAXBException
```

Throws:

```
javax.xml.bind.JAXBException
```

xmlStringToUser

```
public static gov.nasa.eosdis.urs.core.objects.User xmlStringToUser(java.lang.String xml)  
throws javax.xml.bind.JAXBException
```

Throws:

```
javax.xml.bind.JAXBException
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.util

Class UserValidation

```
java.lang.Object
└ gov.nasa.eosdis.urs.util.UserValidation
```

```
public class UserValidation
extends java.lang.Object
```

Constructor Summary

[UserValidation\(\)](#)

[UserValidation\(gov.nasa.eosdis.urs.core.objects.User user\)](#)

[UserValidation\(gov.nasa.eosdis.urs.core.objects.User user, boolean isUpdate\)](#)

[UserValidation\(gov.nasa.eosdis.urs.core.objects.User user, gov.nasa.eosdis.urs.core.common.ChangeAuthority authority\)](#)

Method Summary

| | |
|--|--|
| static java.lang.String | <u>getEmailAddressValidationErrorMessage(java.lang.String email)</u> |
| static java.lang.String | <u>getPasswordValidationErrorMessage(java.lang.String password)</u> |
| static java.lang.String | <u>getPasswordValidationErrorMessage(java.lang.String password, boolean isUpdate)</u> |
| static java.lang.String | <u>getUserNameValidationErrorMessage(java.lang.String uid)</u> |
| java.util.Collection<ValidationErrorMessage> | <u>getValidationErrors()</u> Returns validation errors |
| boolean | <u>isEmailAddressUnique()</u> Tests to see if a user with this email address already exists in URS |
| boolean | <u>isUsernameUnique()</u> Tests to see if a user with this username already exists in URS |
| boolean | <u>isValid()</u> Tests if the user resource is valid and can be expected to succeed when saved. This does not include uniqueness checks, as they generate CONFLICT. |

Overview (User Registration System API 2.55 API)

errors, not validation errors Field Validation Requirements userName cannot be empty, @see gov.nasa.eosdis.urs.core.util.Validation firstName cannot be empty lastName cannot be empty emailAddress cannot be empty, @see gov.nasa.eosdis.urs.core.util.Validation country cannot be empty affiliation cannot be empty accountStatus cannot be empty password cannot be empty, must not be encrypted, configurable by external file @see gov.nasa.eosdis.urs.core.util.Validation portal cannot be empty

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

UserValidation

```
public UserValidation()
```

UserValidation

```
public UserValidation(gov.nasa.eosdis.urs.core.objects.User user)
```

UserValidation

```
public UserValidation(gov.nasa.eosdis.urs.core.objects.User user,  
                      boolean isUpdate)
```

UserValidation

```
public UserValidation(gov.nasa.eosdis.urs.core.objects.User user,  
                      gov.nasa.eosdis.urs.core.common.ChangeAuthority authority)
```

Method Detail

isValid

```
public boolean isValid()
```

Tests if the user resource is valid and can be expected to succeed when saved This does not include uniqueness checks, as they generate CONFLICT errors, not validation errors

Field

Validation Requirements

Overview (User Registration System API 2.55 API)

userName cannot be empty, @see gov.nasa.eosdis.urs.core.util.Validation
firstName cannot be empty
lastName cannot be empty
emailAddress cannot be empty, @see gov.nasa.eosdis.urs.core.util.Validation
country cannot be empty
affiliation cannot be empty
accountStatus cannot be empty
password cannot be empty, must not be encrypted, configurable by external file @see
 gov.nasa.eosdis.urs.core.util.Validation
portal cannot be empty

Returns:

true if the user resource is valid

isUsernameUnique

```
public boolean isUsernameUnique()  
    throws gov.nasa.eosdis.urs.core.dao.exception.DaoException,  
          gov.nasa.eosdis.urs.core.exception.InvalidValueException
```

Tests to see if a user with this username already exists in URS

Returns:

true if a user with this username exists

Throws:

gov.nasa.eosdis.urs.core.dao.exception.DaoException - if unable to connect to LDAP
gov.nasa.eosdis.urs.core.exception.InvalidValueException

isEmailAddressUnique

```
public boolean isEmailAddressUnique()  
    throws gov.nasa.eosdis.urs.core.dao.exception.DaoException,  
          gov.nasa.eosdis.urs.core.exception.InvalidValueException
```

Tests to see if a user with this email address already exists in URS

Returns:

true if a user with this email address exists

Throws:

gov.nasa.eosdis.urs.core.dao.exception.DaoException - if unable to connect to LDAP
gov.nasa.eosdis.urs.core.exception.InvalidValueException

getValidationErrors

```
public java.util.Collection<ValidationError> getValidationErrors()
```

Returns validation errors

Returns:

a mapping of errors to fields, or an empty hash if the resource is valid

getUserNameValidationError

```
public static java.lang.String getUserNameValidationError(java.lang.String uid)
```

getEmailAddressValidationError

```
public static java.lang.String getEmailAddressValidationError(java.lang.String email)
```

getPasswordValidationError

```
public static java.lang.String getPasswordValidationError(java.lang.String password,  
                           boolean isUpdate)
```

getPasswordValidationError

```
public static java.lang.String getPasswordValidationError(java.lang.String password)
```

Copyright © 2012. All Rights Reserved.

gov.nasa.eosdis.urs.webapp Class UrsServletContextListener

```
java.lang.Object
└ gov.nasa.eosdis.urs.webapp.UrsServletContextListener
```

All Implemented Interfaces:

```
java.util.EventListener, javax.servlet.ServletContextListener
```

```
public class UrsServletContextListener
extends java.lang.Object
implements javax.servlet.ServletContextListener
```

Constructor Summary

| | |
|--|--|
| <u>UrsServletContextListener()</u> | |
|--|--|

Method Summary

| | |
|------|---|
| void | <u>contextDestroyed(javax.servlet.ServletContextEvent arg0)</u> |
| void | <u>contextInitialized(javax.servlet.ServletContextEvent arg0)</u> |

Methods inherited from class java.lang.Object

| |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |
|---|

Constructor Detail

UrsServletContextListener

```
public UrsServletContextListener()
```

Method Detail

contextDestroyed

```
public void contextDestroyed(javax.servlet.ServletContextEvent arg0)
```

Specified by:

contextDestroyed in interface javax.servlet.ServletContextListener

contextInitialized

```
public void contextInitialized(javax.servlet.ServletContextEvent arg0)
```

Specified by:

contextInitialized in interface javax.servlet.ServletContextListener

Copyright © 2012. All Rights Reserved.

Constant Field Values

Contents

- [gov.nasa.*](#)

gov.nasa.*

gov.nasa.eosdis.urs.util.AccountVerifier

| | | |
|--------------------------------------|---------------------|----------------|
| public static final java.lang.String | <u>ACTIVATION</u> | "activation" |
| public static final java.lang.String | <u>VERIFICATION</u> | "verification" |

gov.nasa.eosdis.urs.util.StringConstants

| | | |
|--------------------------------------|------------------|--------------------------------|
| public static final java.lang.String | <u>DIGITS</u> | "0123456789" |
| public static final java.lang.String | <u>HEX_UPPER</u> | "0123456789ABCDEF" |
| public static final java.lang.String | <u>PUNCT</u> | "!@#\$%^&*()_=+.,<>/?\\ [{ }]" |
| public static final java.lang.String | <u>UPPER</u> | "ABCDEFGHIJKLMNPQRSTUVWXYZ" |

Copyright © 2012. All Rights Reserved.

Deprecated API

Contents

Copyright © 2012. All Rights Reserved.

A B C D E F G H I J L M N P R S T U V X

A

[AccountVerifier](#) - Class in [gov.nasa.eosdis.urs.util](#)

[AccountVerifier\(User, String\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

Class constructor.

[AccountVerifier\(User, String, String\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

Class constructor.

[AccountVerifier\(String, String\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

Class constructor.

[activateAccount\(String\)](#) - Method in class gov.nasa.eosdis.urs.api.[RpcResource](#)

Account activation method.

[ACTIVATION](#) - Static variable in class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

[addUserToGroup\(String, String, String\)](#) - Method in class gov.nasa.eosdis.urs.api.[GroupResource](#)

Adds a user to a group.

[ALNUM](#) - Static variable in class gov.nasa.eosdis.urs.util.[StringConstants](#)

The set of letters and numbers.

[ALPHA](#) - Static variable in class gov.nasa.eosdis.urs.util.[StringConstants](#)

The set of upper- and lowercase letters.

[AuthorityValidation](#) - Class in [gov.nasa.eosdis.urs.util](#)

[AuthorityValidation\(ChangeAuthority\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[AuthorityValidation](#)

B

[BadRequestException](#) - Exception in [gov.nasa.eosdis.urs.api.exceptions](#)

[BadRequestException\(String\)](#) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.[BadRequestException](#)

C

[choose\(T...\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Randomly selects an item from an array.

[choose\(List<T>\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Randomly selects an item from a list.

[ConflictErrorException](#) - Exception in [gov.nasa.eosdis.urs.api.exceptions](#)

[ConflictErrorException\(String\)](#) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.[ConflictErrorException](#)

[contextDestroyed\(ServletContextEvent\)](#) - Method in class gov.nasa.eosdis.urs.webapp.[UrsApiServletContextListener](#)

Overview (User Registration System API 2.55 API)

contextInitialized(ServletContextEvent) - Method in class gov.nasa.eosdis.urs.webapp.UrsApiServletContextListener

createGroup(String, Group) - Method in class gov.nasa.eosdis.urs.api.GroupResource

Creates a named group.

createUser(String, String, boolean, User) - Method in class gov.nasa.eosdis.urs.api.UserResource

User 'Create' operation.

D

deleteGroup(String, String) - Method in class gov.nasa.eosdis.urs.api.GroupResource

Deletes a group from the database.

deleteMail(String) - Method in class gov.nasa.eosdis.urs.api.TestResource

DELETE method for performing the deletion of a mail file in the mail drop box.

deleteUser(String, String) - Method in class gov.nasa.eosdis.urs.api.UserResource

User 'delete' operation.

deleteUserFromGroup(String, String, String) - Method in class gov.nasa.eosdis.urs.api.GroupResource

Delete user from group.

DIGITS - Static variable in class gov.nasa.eosdis.urs.util.StringConstants

The set of numeric characters.

F

field - Variable in class gov.nasa.eosdis.urs.api.objects.ValidationError

findUser(UserDAO, ChangeAuthority, String) - Static method in class gov.nasa.eosdis.urs.util.UserUtils

Finds the user in LDAP by user name (uid).

findUserByEmail(UserDAO, ChangeAuthority, String) - Static method in class gov.nasa.eosdis.urs.util.UserUtils

Finds a user in LDAP by email address.

ForbiddenException - Exception in gov.nasa.eosdis.urs.api.exceptions

ForbiddenException(String) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.ForbiddenException

ForbiddenException() - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.ForbiddenException

G

generateMail(String, HashMap<String, String>) - Static method in class gov.nasa.eosdis.urs.util.Mailer

Generates an email body from a named resource file and a series of parameters.

getAffiliations() - Method in class gov.nasa.eosdis.urs.api.ValidsResource

GET method for requesting a list of acceptable (valid) affiliation values.

GetAllAppAuthenticationMetrics(String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.MetricResource

Get all application authentication metrics.

GetAllUserAuthenticationMetrics(String, String, String, String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.MetricResource

Get all user authentication metrics.

getApp() - Method in class gov.nasa.eosdis.urs.util.AccountVerifier

Overview (User Registration System API 2.55 API)

Returns the application that requested creation of this account.

GetAppAuthenticationMetrics(String, String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.MetricResource

Get a specific application authentication metrics.

GetAuthenticationMetrics(String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.MetricResource

Get general authentication metrics.

getCountries() - Method in class gov.nasa.eosdis.urs.api.ValidsResource

GET method for requesting a list of acceptable (valid) countries.

getCustomAppContent(String, String, String, List<String>) - Method in class gov.nasa.eosdis.urs.api.RpcResource

The methods is used by applications to retrieve application specific data.

getEmailAddressValidationError(String) - Static method in class gov.nasa.eosdis.urs.util.UserValidation

getGroup(String, String) - Method in class gov.nasa.eosdis.urs.api.GroupResource

Returns all the details for a named group.

getGroups(String) - Method in class gov.nasa.eosdis.urs.api.GroupResource

Returns a list of the groups available for use.

getKey() - Method in class gov.nasa.eosdis.urs.util.AccountVerifier

Returns the activation key for this account.

getMail(String, String, String) - Method in class gov.nasa.eosdis.urs.api.TestResource

GET method which returns all the mail files in the current inbox, with an optional query parameters.

getMessage() - Method in exception gov.nasa.eosdis.urs.api.exceptions.BadRequestException

getMessage() - Method in exception gov.nasa.eosdis.urs.api.exceptions.ConflictErrorException

getMessage() - Method in exception gov.nasa.eosdis.urs.api.exceptions.ValidationErrorsException

getPartialUser(User, List<String>) - Static method in class gov.nasa.eosdis.urs.util.UserUtils

Creates a partial user object from a full user object, using a list of desired user attributes.

getPassword() - Method in class gov.nasa.eosdis.urs.api.objects.LoginRequest

getPasswordValidationError(String, boolean) - Static method in class gov.nasa.eosdis.urs.util.UserValidation

getPasswordValidationError(String) - Static method in class gov.nasa.eosdis.urs.util.UserValidation

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.BadRequestException

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.ConflictErrorException

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.ForbiddenException

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.InternalServerError

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.ServiceUnavailableException

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.UnauthorizedException

getResponse() - Method in exception gov.nasa.eosdis.urs.api.exceptions.ValidationErrorsException

getStudyAreas() - Method in class gov.nasa.eosdis.urs.api.ValidsResource

GET method for requesting a list of acceptable (valid) primary study areas.

getUser(String, String, List<String>) - Method in class gov.nasa.eosdis.urs.api.UserResource

User 'Get' operation.

Overview (User Registration System API 2.55 API)

getUser() - Method in class gov.nasa.eosdis.urs.util.AccountVerifier

Returns the user account details.

GetUserAuthenticationMetrics(String, String, String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.MetricResource

Get the specified user authentication metrics.

getUserFields() - Method in class gov.nasa.eosdis.urs.api.objects.LoginRequest

getUsername() - Method in class gov.nasa.eosdis.urs.api.objects.LoginRequest

getUserNameValidationError(String) - Static method in class gov.nasa.eosdis.urs.util.UserValidation

getUsers(String, String, String, String, String, String, String, String, String, String, int, int, boolean, boolean, String) - Method in class gov.nasa.eosdis.urs.api.UserResource

Search users by a variety of fields.

GetUsersCountsFromLDAP(String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.MetricResource

Endpoint that gets the number of new users or total users.

getUserTypes() - Method in class gov.nasa.eosdis.urs.api.ValidsResource

GET method for requesting a list of acceptable (valid) user types.

getValidationErrors() - Method in class gov.nasa.eosdis.urs.util.UserValidation

Returns validation errors

getVersion() - Method in class gov.nasa.eosdis.urs.api.RpcResource

GET method to get the version of the URS API.

gov.nasa.eosdis.urs.api - package gov.nasa.eosdis.urs.api

gov.nasa.eosdis.urs.api.exceptions - package gov.nasa.eosdis.urs.api.exceptions

gov.nasa.eosdis.urs.api.objects - package gov.nasa.eosdis.urs.api.objects

gov.nasa.eosdis.urs.extendedwadl - package gov.nasa.eosdis.urs.extendedwadl

gov.nasa.eosdis.urs.util - package gov.nasa.eosdis.urs.util

gov.nasa.eosdis.urs.webapp - package gov.nasa.eosdis.urs.webapp

GroupList - Class in gov.nasa.eosdis.urs.api.objects

GroupList() - Constructor for class gov.nasa.eosdis.urs.api.objects.GroupList

Class constructor.

GroupList(List<GroupInfo>) - Constructor for class gov.nasa.eosdis.urs.api.objects.GroupList

Class constructor.

GroupResource - Class in gov.nasa.eosdis.urs.api

Server side REST interface for performing group operations.

GroupResource() - Constructor for class gov.nasa.eosdis.urs.api.GroupResource

groups - Variable in class gov.nasa.eosdis.urs.api.objects.GroupList

GroupUtils - Class in gov.nasa.eosdis.urs.util

GroupUtils() - Constructor for class gov.nasa.eosdis.urs.util.GroupUtils

H

HEX_LOWER - Static variable in class gov.nasa.eosdis.urs.util.StringConstants

Lowercase hexadecimal digits.

HEX_UPPER - Static variable in class gov.nasa.eosdis.urs.util.StringConstants

Uppercase hexadecimal digits.

I

InternalServerError - Exception in gov.nasa.eosdis.urs.api.exceptions

InternalServerError() - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.InternalServerError

InternalServerError(String) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.InternalServerError

InternalServerError(Exception) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.InternalServerError

InternalServerError(String, Exception) - Constructor for exception

gov.nasa.eosdis.urs.api.exceptions.InternalServerError

isAuthorityValid() - Method in class gov.nasa.eosdis.urs.util.AuthorityValidation

isEmailAddressUnique() - Method in class gov.nasa.eosdis.urs.util.UserValidation

Tests to see if a user with this email address already exists in URS

isPasswordEncrypted(String) - Static method in class gov.nasa.eosdis.urs.util.PasswordUtils

Checks to see whether a user password is encrypted

isPasswordPlaintext(String) - Static method in class gov.nasa.eosdis.urs.util.PasswordUtils

Verifies that the password does not have an encryption prefix.

isUsernameUnique() - Method in class gov.nasa.eosdis.urs.util.UserValidation

Tests to see if a user with this username already exists in URS

isValid() - Method in class gov.nasa.eosdis.urs.util.UserValidation

Tests if the user resource is valid and can be expected to succeed when saved This does not include uniqueness checks, as they generate CONFLICT errors, not validation errors Field Validation Requirements userName cannot be empty, @see gov.nasa.eosdis.urs.core.util.Validation firstName cannot be empty lastName cannot be empty emailAddress cannot be empty, @see gov.nasa.eosdis.urs.core.util.Validation country cannot be empty affiliation cannot be empty accountStatus cannot be empty password cannot be empty, must not be encrypted, configurable by external file @see gov.nasa.eosdis.urs.core.util.Validation portal cannot be empty

isValidGroupName(String) - Static method in class gov.nasa.eosdis.urs.util.GroupUtils

Determines if a given group name is valid

J

jaxbObjectToXmlString(Object) - Static method in class gov.nasa.eosdis.urs.util.UserUtils

L

[**loadResource\(String\)**](#) - Static method in class gov.nasa.eosdis.urs.util.ResourceLoader

Loads a named resource file.

[**loadResourceList\(String\)**](#) - Static method in class gov.nasa.eosdis.urs.util.ResourceLoader

Loads a named resource file.

[**LoginRequest**](#) - Class in gov.nasa.eosdis.urs.api.objects

[**LoginRequest\(\)**](#) - Constructor for class gov.nasa.eosdis.urs.api.objects.LoginRequest

[**LoginRequest\(String, String\)**](#) - Constructor for class gov.nasa.eosdis.urs.api.objects.LoginRequest

[**LoginRequest\(String, String, UserFields\)**](#) - Constructor for class gov.nasa.eosdis.urs.api.objects.LoginRequest

[**LoginResource**](#) - Class in gov.nasa.eosdis.urs.api

Server side REST interface for performing user authentication operations.

This class handles the logins to URS for other applications.

[**LoginResource\(\)**](#) - Constructor for class gov.nasa.eosdis.urs.api.LoginResource

[**LOWER**](#) - Static variable in class gov.nasa.eosdis.urs.util.StringConstants

The set of lowercase letters.

M

[**Mailer**](#) - Class in gov.nasa.eosdis.urs.util

Utility class used to compose emails for sending.

[**Mailer\(\)**](#) - Constructor for class gov.nasa.eosdis.urs.util.Mailer

[**main\(String\[\]\)**](#) - Static method in class gov.nasa.eosdis.urs.util.Randomizer

[**message**](#) - Variable in class gov.nasa.eosdis.urs.api.objects.ValidationError

[**MetricResource**](#) - Class in gov.nasa.eosdis.urs.api

This class defines methods that can be used to retrieve new users created in LDAP and authentication metrics given a date time range.

[**MetricResource\(\)**](#) - Constructor for class gov.nasa.eosdis.urs.api.MetricResource

[**mockCreateUser\(String, String, boolean, boolean, User\)**](#) - Method in class gov.nasa.eosdis.urs.api.TestResource

POST method used to mock user creation operation.

N

[**nextBoolean\(\)**](#) - Method in class gov.nasa.eosdis.urs.util.Randomizer

[**nextBytes\(byte\[\]\)**](#) - Method in class gov.nasa.eosdis.urs.util.Randomizer

Generates random bytes and places them into a user-supplied byte array.

[**nextDouble\(\)**](#) - Method in class gov.nasa.eosdis.urs.util.Randomizer

[nextFloat\(\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

[nextGaussian\(\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

[nextInt\(\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

[nextInt\(int\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

[nextInt\(int, int\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

 Returns a randomly chosen int between 'min' inclusive and 'max' exclusive.

[nextLong\(\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

[nextLong\(long\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

P

[PasswordUtils](#) - Class in gov.nasa.eosdis.urs.util

[PasswordUtils\(\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[PasswordUtils](#)

[postLogin\(String, String, LoginRequest\)](#) - Method in class gov.nasa.eosdis.urs.api.[LoginResource](#)

 POST method for performing a user authentication.

[postLoginUrlEncoded\(String, String, String, String\)](#) - Method in class gov.nasa.eosdis.urs.api.[LoginResource](#)

 POST method for performing a login, accepting data of type "application/x-www-form-urlencoded".

[PUNCT](#) - Static variable in class gov.nasa.eosdis.urs.util.[StringConstants](#)

 Punctuation characters.

R

[random\(String, int\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

 Creates a String of 'length' characters randomly chosen from 'source'.

[random\(String, int, int\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

 Randomly selects characters from 'source' to build a new String whose length is between 'min' and 'max', inclusive.

[Randomizer](#) - Class in gov.nasa.eosdis.urs.util

 Randomizer is a pseudo random number generator that builds on java.util.Random and allows multiple instances to share a single instance of that class, which should be more random than creating a new Random instance every time one is needed.

[Randomizer\(\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[Randomizer](#)

 Creates a new Randomizer instance that uses the singleton random number generator shared by all other Randomizer instances created with this constructor.

[Randomizer\(long\)](#) - Constructor for class gov.nasa.eosdis.urs.util.[Randomizer](#)

 Creates a new Randomizer with the given seed for repeatable pseudorandomness.

[randomPassword\(int, int\)](#) - Static method in class gov.nasa.eosdis.urs.util.[PasswordUtils](#)

 Generates a random password with upper and lower constraints on its length.

[randomPassword\(int\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

 Generates a password of upper and lowercase letters and digits, where each character is equally likely to be from each of the 3 character sets.

[randomPassword\(int, String...\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Overview (User Registration System API 2.55 API)

Generates a random string from the given character sets, so that each character is equally likely to come from any character set, resulting in a different distribution than randomly choosing characters from one large character set.

resetPassword(String, String) - Method in class gov.nasa.eosdis.urs.api.[RpcResource](#)

POST method for performing a password reset for a user.

ResourceLoader - Class in [gov.nasa.eosdis.urs.util](#)

ResourceLoader() - Constructor for class gov.nasa.eosdis.urs.util.[ResourceLoader](#)

retrieveUsername(String, String) - Method in class gov.nasa.eosdis.urs.api.[RpcResource](#)

Username reminder method.

RpcResource - Class in [gov.nasa.eosdis.urs.api](#)

Server side REST interface for performing RPC operations.

RpcResource() - Constructor for class gov.nasa.eosdis.urs.api.[RpcResource](#)

S

SAMPLE GROUP - Static variable in class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

SAMPLE LOGIN REQUEST - Static variable in class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

SAMPLE SEARCH RESPONSE - Static variable in class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

SAMPLE USER - Static variable in class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

SAMPLE VALIDATION ERROR - Static variable in class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

SAMPLE VALIDATION ERRORS - Static variable in class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

sanityCheck(String) - Method in class gov.nasa.eosdis.urs.api.[LoginResource](#)

Sanity check.

send(String, String, String, String) - Static method in class gov.nasa.eosdis.urs.util.[Mailer](#)

Writes an email into a file for processing by the mailer script.

sendMail(String, String, String, String) - Method in class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

Sends an email with an account activation link embedded into it.

sendMail(String, String, String) - Method in class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

Send mail.

ServiceUnavailableException - Exception in [gov.nasa.eosdis.urs.api.exceptions](#)

ServiceUnavailableException() - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.[ServiceUnavailableException](#)

ServiceUnavailableException(String) - Constructor for exception

gov.nasa.eosdis.urs.api.exceptions.[ServiceUnavailableException](#)

setCustomAppContent(String, String, String, String, String) - Method in class gov.nasa.eosdis.urs.api.[RpcResource](#)

This method sets the customAppContent attribute of a specified user in LDAP.

setPassword(String) - Method in class gov.nasa.eosdis.urs.api.objects.[LoginRequest](#)

setUserFields(UserFields) - Method in class gov.nasa.eosdis.urs.api.objects.[LoginRequest](#)

[setUsername\(String\)](#) - Method in class gov.nasa.eosdis.urs.api.objects.[LoginRequest](#)

[shuffle\(List<?>\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Randomly permutes a list destructively.

[shuffle\(T\[\]\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Randomly permutes an array destructively.

[shuffled\(List<T>\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Randomly permutes a copy of list.

[shuffled\(T\[\]\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Randomly permutes a copy of array.

[shuffled\(String\)](#) - Method in class gov.nasa.eosdis.urs.util.[Randomizer](#)

Returns a randomly permuted copy of str.

[StringConstants](#) - Class in [gov.nasa.eosdis.urs.util](#)

T

[TestResource](#) - Class in [gov.nasa.eosdis.urs.api](#)

Tools and utilities used only for testing purposes.

This class is not deployed in the operational system.

[TestResource\(\)](#) - Constructor for class gov.nasa.eosdis.urs.api.[TestResource](#)

[toString\(\)](#) - Method in class gov.nasa.eosdis.urs.api.objects.[LoginRequest](#)

U

[UnauthorizedException](#) - Exception in [gov.nasa.eosdis.urs.api.exceptions](#)

Exception to support basic http authentication.

[UnauthorizedException\(\)](#) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.[UnauthorizedException](#)

[UnauthorizedException\(String\)](#) - Constructor for exception gov.nasa.eosdis.urs.api.exceptions.[UnauthorizedException](#)

[updateGroupUsers\(String, String, Group\)](#) - Method in class gov.nasa.eosdis.urs.api.[GroupResource](#)

Update group's users.

[updateUser\(String, String, boolean, String, User\)](#) - Method in class gov.nasa.eosdis.urs.api.[UserResource](#)

User 'Update' operation.

[UPPER](#) - Static variable in class gov.nasa.eosdis.urs.util.[StringConstants](#)

String classes for generating values composed of specific sets of characters (e.g., all letters, all numbers, hex digits).

[UrsApiServletContextListener](#) - Class in [gov.nasa.eosdis.urs.webapp](#)

[UrsApiServletContextListener\(\)](#) - Constructor for class gov.nasa.eosdis.urs.webapp.[UrsApiServletContextListener](#)

[URSBaseResource](#) - Class in [gov.nasa.eosdis.urs.api](#)

The base class URSBaseResource is the super class of classes GroupResource, LoginResource, RpcResource, TestResource and UserResource.

[URSBaseResource\(\)](#) - Constructor for class gov.nasa.eosdis.urs.api.[URSBaseResource](#)

[URSModelExamples](#) - Class in [gov.nasa.eosdis.urs.extendedwadl](#)

Overview (User Registration System API 2.55 API)

This class provides example representations that can be used for wadl docs.

URSModelExamples() - Constructor for class gov.nasa.eosdis.urs.extendedwadl.[URSModelExamples](#)

UserResource - Class in [gov.nasa.eosdis.urs.api](#)

Server side REST interface for performing user based operations, such as create, read, update, and delete.

UserResource() - Constructor for class gov.nasa.eosdis.urs.api.[UserResource](#)

UserUtils - Class in [gov.nasa.eosdis.urs.util](#)

UserUtils() - Constructor for class gov.nasa.eosdis.urs.util.[UserUtils](#)

UserValidation - Class in [gov.nasa.eosdis.urs.util](#)

UserValidation() - Constructor for class gov.nasa.eosdis.urs.util.[UserValidation](#)

UserValidation(User) - Constructor for class gov.nasa.eosdis.urs.util.[UserValidation](#)

UserValidation(User, boolean) - Constructor for class gov.nasa.eosdis.urs.util.[UserValidation](#)

UserValidation(User, ChangeAuthority) - Constructor for class gov.nasa.eosdis.urs.util.[UserValidation](#)

UserValidationResource - Class in [gov.nasa.eosdis.urs.api](#)

Server-side REST interface for validating various components of a user before creation of or updates to that user.

UserValidationResource() - Constructor for class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

V

validateEmailFromQuery(String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a user's email address.

validateEmailFromUser(String, String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a submitted user's email address (usually for updating).

validatePasswordFromQuery(String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a user's password.

validatePasswordFromUser(String, String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a submitted user's password (usually for updating).

validateUpdatedUser(String, String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a submitted user resource (usually for updating).

validateUser(String, String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a submitted user resource.

validateUserAlias(String, String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a submitted user resource.

validateUsernameFromQuery(String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a user's username.

validateUsernameFromUser(String, String) - Method in class gov.nasa.eosdis.urs.api.[UserValidationResource](#)

POST method for validating a submitted user's username (usually for updating).

ValidationError - Class in [gov.nasa.eosdis.urs.api.objects](#)

ValidationError() - Constructor for class gov.nasa.eosdis.urs.api.objects.[ValidationError](#)

[ValidationError\(String, String\)](#) - Constructor for class gov.nasa.eosdis.urs.api.objects.[ValidationError](#)

[ValidationErrors](#) - Class in [gov.nasa.eosdis.urs.api.objects](#)

[ValidationErrors\(\)](#) - Constructor for class gov.nasa.eosdis.urs.api.objects.[ValidationErrors](#)

[ValidationErrors\(Collection<ValidationError>\)](#) - Constructor for class gov.nasa.eosdis.urs.api.objects.[ValidationErrors](#)

[validationErrors](#) - Variable in class gov.nasa.eosdis.urs.api.objects.[ValidationErrors](#)

[ValidationErrorsException](#) - Exception in [gov.nasa.eosdis.urs.api.exceptions](#)

[ValidationErrorsException\(ValidationErrors\)](#) - Constructor for exception
gov.nasa.eosdis.urs.api.exceptions.[ValidationErrorsException](#)

[ValidationErrorsException\(Collection<ValidationError>\)](#) - Constructor for exception
gov.nasa.eosdis.urs.api.exceptions.[ValidationErrorsException](#)

[ValidationErrorsException\(String, String\)](#) - Constructor for exception
gov.nasa.eosdis.urs.api.exceptions.[ValidationErrorsException](#)

[ValidsResource](#) - Class in [gov.nasa.eosdis.urs.api](#)

Server side REST interface for requesting lists of valid values for a given property.

[ValidsResource\(\)](#) - Constructor for class gov.nasa.eosdis.urs.api.[ValidsResource](#)

[VERIFICATION](#) - Static variable in class gov.nasa.eosdis.urs.util.[AccountVerifier](#)

[verifyAccount\(String\)](#) - Method in class gov.nasa.eosdis.urs.api.[RpcResource](#)

Account verification method.

X

[xmlStringToUser\(String\)](#) - Static method in class gov.nasa.eosdis.urs.util.[UserUtils](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [L](#) [M](#) [N](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#) [X](#)

Copyright © 2012. All Rights Reserved.